HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences

Kurt Baarman

Thermal Characterization of Vertically Stacked Chips

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics.

Espoo, March 28, 2008

Supervisor: Prof. Rolf Stenberg
Instructor: Dr. Tech. Mikko Lyly

HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Kurt Baarman
Degree Programme:	Degree Programme in Engineering Physics
Major subject:	Physics
Minor subject:	Theoretical and Applied Mechanics
Title:	Thermal Characterization of Vertically Stacked Chips
Title in Swedish:	Termisk karakterisering av staplade mikrokretsar
Chair:	Mat-5, Theoretical and Applied Mechanics
Supervisor:	Prof. Rolf Stenberg
Instructor:	Dr. Tech. Mikko Lyly

Abstract:
In this master's thesis, we review several thermal management solutions for stacked chips in portable electronics. We simulate the cooling benefits of potential thermal management solutions. Furthermore, we investigate the effect of stack layout, stacking order, and distributed power dissipation regions on the temperature.

We design a chip stack based on the design guidelines discovered in the initial investigation. Then we calculate the in-stack thermal strain induced by the temperature distribution for this stack.

We conclude that proper stack design can provide considerable reduction of maximum stack temperature. While the maximum temperature of a 2 W-stack designed without consideration of thermal management can exceed 95 °C, the maximum in-stack operational temperature is below 85 °C for the stack design in the Nokia Virtual Thermal Test Environment.

Number of pages: 66 **Keywords:** chip stack, electronics cooling, finite element, heat load personal electronics, thermal management, thermal strain

Faculty fills
Approved: **Library code:**

TEKNISKA HÖGSKOLAN
Fakulteten för Informations- och Naturvetenskaper

SAMMANDRAG AV DIPLOMARBETE

Utfört av:	Kurt Baarman
Utbildningsprogram:	Utbildningsprogrammet för Teknisk Fysik
Huvudämne:	Fysik
Biämne:	Mekanik

Arbetets namn:	Termisk karakterisering av staplade mikrokretsar
Title in English:	Thermal Characterization of Vertically Stacked Chips
Professur:	Mat-5, Mekanik
Övervakare:	Prof. Rolf Stenberg
Handledare:	TkD Mikko Lyly

Sammandrag:
I detta diplomarbete granskar vi olika möjligheter för kylning av staplade mikrokretsar i bärbar elektronik. Vi simulerar verkan av kylmetoder som skulle kunna användas för termisk reglering. Vidare undersöker vi hur stapelns arkitektur samt effektfördelning påverkar temparaturen.

Baserat på våra simuleringsresultat planerar vi arkitekturen för en staplad mikrokrets. För denna krets beräknar vi även den termiska spänningen i stapeln som förorsakas av värmefördelningen.

Vi drar slutsatsen att den maximala temperaturen i en välplanerad stapel av mikrokretsar är betydligt lägre än i en stapel vars arkitektur inte har blivit planerad med tanke på termisk reglering. I det senare fallet kan temperaturen överskrida 95 °C för en 2 W:s stapel, medan temperaturen för den planerade stapelarkitekturen underskrider 85 °C i den använda modellen.

Sidoantal: 66

Nyckelord: bärbar elektronik, finita element, staplad mikrokrets, termisk last, termisk reglering, termisk spänning

Ifylles på avdelningen

Godkänd: **Bibliotek:**

Acknowledgments

I want to thank my supervisor, Professor R. Stenberg, who reminded me where physics stops and linear algebra begins. I am grateful to my instructor, Doctor M. Lyly. Without the introduction to Elmer provided by him, getting started would have been considerably more difficult.

Professor V. Havu has given me valuable comments on this thesis, and helped me out when I was short on symbols. Only the letters o and r remain unassigned. I also want to thank him, and the rest of the crew in the mechanics' coffee room, for a pleasant working environment.

Moreover, I want to thank the entire PICO group, especially Professor J. Pekola and Doctor M. Meschke, for discussions over the years, most recently on in-chip thin film thermometry.

Doctor P. Råback has provided insightful advice and new functionality for the Elmer software. Thank you.

Further, I want to thank Docent S. Franssila for advice on small scale cooling solutions.

The work documented in this master's thesis was funded by Nokia's SAUNA project. I especially want to thank Doctor M. Kuulusa.

Part of this master's thesis was done at CSC, the Finnish IT Center for Science, in Keilaniemi during the period May 21 to September 14, 2007. The multiphysics simulations have also been run on CSC's Murska cluster. This thesis was finished at the Institute of Mathematics at TKK.

Furthermore, I want to thank my parents and my brother for their support. Finally, I want to thank Piia, for making me write what I meant to say, and for making it all endurable.

Kurt Baarman
Espoo, March 28, 2008

Contents

Chapter 1

Introduction

Nowadays, personal electronics are expected to simultaneously provide several user applications. Therefore, development is driven towards handsets with more processing power. The increased output power increases the need to manage handset temperature. However, while large and heavy thermal management solutions can be used in personal computers, handset thermal management solutions are severely limited by size and weight.

Due to limited space, one way to meet the growing processing power requirements of handsets is to vertically integrate individual chips and packages. The increased vertical integration results in a high power dissipation density in packages. Thermal management becomes critical when die stacks contain dies with a low maximum allowed operational temperature. Thermal characterization of vertically stacked die packages is, therefore, an important part of package design.

Characterization is often done by computer simulation. Computational methods are used to investigate thermal behavior of die stacks in greater detail than what is possible with experimental thermometry. Furthermore, simulations offer the chance to test future technological solutions.

In classical mechanics, heat transfer is composed of three separate effects: conduction through a material, radiation from a surface, and convection with a moving medium. Convection is only present in fluids. It is the transport of heat bound to a volume along with the moving fluid. Thus, convection can be regarded not as an independent heat transfer effect, but as a result of mass transfer. Mass transfer takes place due to an external force, *e.g.* a fan, or is induced naturally. In particular, when air is heated, it expands and rises above colder, denser air. The rising hot air is replaced by cooler air, effectively transporting heat away from the heat source. In the confined space of a handset with comparatively small thermal gradients, heat transfer is dominated by conduction.

In this master's thesis, we investigate the thermal behavior of vertical chip stacks in the Nokia Virtual Thermal Test Environment Version 4. The

finite element model is implemented in Elmer. Mesh generation and description of the model geometry is done with Gmsh.

We begin by presenting thermal management solutions in Chap. 2. Chapter 3 presents the physics required for thermal characterization of the chip stacks. Chapter 4 deals with the mathematical tools necessary for assembly and solution of the linear system of equations representing the physical problem. We present the stack layout, thermal test environment, and software used to calculate the thermal distribution in Chap. 5. Chapter 6 presents the results obtained, and in Chap. 7 we draw conclusions based on the earlier results and present a starting point for experimental verification of the results and the computational environment.

Chapter 2

Thermal management in handsets

Due to limited space and weight of handsets, thermal management solutions must be small and light. Thermal management solutions can be divided into three different groups: active coolers, passive coolers, and phase change materials (PCM).

In handsets, active cooling by forced convection is restricted by the device size and by the difficulty of maintaining the necessary heat bath for the moving air mass. Therefore, active cooling solutions are limited to microelectromechanical systems (MEMS). Proposed MEMS cooling solutions include thermoelectric heat pumps and microchannel (MC) liquid coolers.

2.1 Microchannel liquid cooling

MC cooling of electronics was introduced more than 20 years ago. Although significant cooling has been achieved with this method, practical implementation currently confronts a number of challenges. On an application level, conflicting requirements are placed on the cooling liquid, and severe demands are placed on the micropumps. More critically, we are unaware of techniques for reliable assembly of MC flow networks across adhesive interfaces. For this reason, we do not consider MC cooling an attractive cooling solution in the short term [1–5].

In order to avoid frequent maintenance, the entire MC flow network must be hermetically sealed, and the cooling liquid must provide lubrication for mechanical pumps. While a liquid with high viscosity would be preferable as lubricant, a low viscosity liquid would place less severe requirements on the micropump. In addition, the coolant should also have a high thermal conductivity, be non-poisonous, and withstand standard freezing and heating requirements for electronics. A flow that is driven by a magnetic or an electric field can be fabricated without mechanical pumps, permitting lubrication

requirements to be disregarded [4,6].

In principle, the coolant should not damage surrounding electronics in the case of a leak. However, if the operation of the stack depends on MC cooling, device breakdown in the case of coolant escape can not be prevented by the use of a non-damaging coolant.

Heat transport in MC coolers can be increased by the use of a phase changing fluid. However, poor flow distribution in two-phase flow can lead to local dry out and massive temperature fluctuations on the order of 30 °C for 600 mW hotspot heating or fluctuations on the order of 20 °C for 400 mW hotspot heating. Even single-phase MC flow is capable of cooling very high heat flux hot spots. Cooling in excess of 790 W/cm^2 has been demonstrated [5,6].

It would be particularly efficient to use MC heat transport to lower the thermal resistances of critical interface layers. Other dissipation techniques could then be used to transfer heat further into the heat bath.

It is difficult to fabricate a MC flow network bridging the package-board and the package-heatspreader interface. It is unclear how the package assembly of the interface crossing microchannels could be made reliable. Congestion of the microchannels would render the entire cooling system non-operational.

2.2 Thermoelectric cooling

Thin film thermoelectric coolers (TFTEC) are also presented as a local cooling solution. While the hotspot is cooled, the TFTEC will lead to an increased burden on other cooling components due to the additional power consumed by the TFTEC. Therefore, TFTECs are particularly efficient at cooling a single, concentrated hotspot. When considering a non-ideal TFTEC, thermal contact resistance of the device can cause the hotspot temperature to actually increase [6].

In mobile handset processors, the per chip heat production is around 200 mW, while processors in personal computers can easily exceed 100 W. High power dissipation in die stacks originates from the vertical integration of multiple chips. This results in a comparatively uniform temperature distribution. Consequently, thermoelectric coolers will not be particularly effective at protecting die stacks from overheating.

2.3 Enhanced thermal conductivity

Passive cooling by decreased adhesive and interface material thermal resistance is one of the most straightforward ways of decreasing the operational temperature of a device. Particle laden interface materials work by increasing the volume fraction of high conductivity micro- or nanoparticles until the

percolation threshold is reached. Small aluminum spheres or nanotubes are the most important laden particle types. Typical proposed particle volume fractions are $40 - 60$ % [6–10].

Particle lading also changes the mechanical properties of the interface layer. In particular, an increased particle volume fraction results in an increased boundary layer thickness and an increased thermal contact resistance. Figure 2.1 presents a schematic origin of the different terms in the thermal resistance of an interface layer. When the volume fraction of solid particles exceeds a certain threshold, the effective thermal resistance of the interface layer starts to increase regardless of the laden material [9].

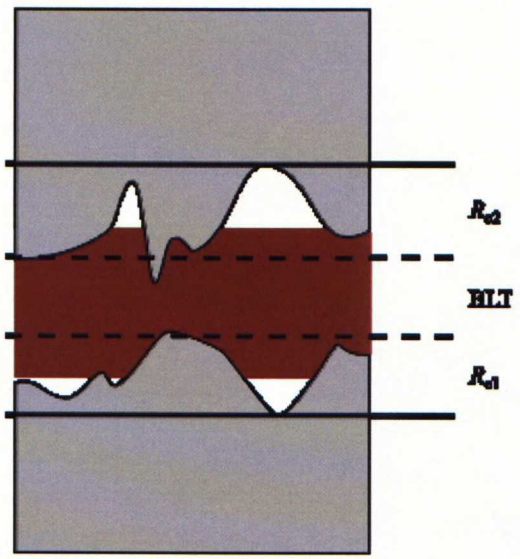

Figure 2.1: Schematic figure of the different thermal resistance sources for a interfacial adhesive layer. For very thin boundary layers the contact resistances, R_{c1} and R_{c2}, dominate.

Even though the increased thermal conductivity of interface materials currently offers the most promising method for cooling of vertically stacked chips, their importance will diminish in the long run. With increased stacking requirements, both the die thickness and the bond layer thickness are decreasing. Therefore, contact resistance of the interface will play an increasingly dominant role in determination of the thermal resistance of an interface layer.

Although graphene has very impressive thermal conductivity properties, there are currently no practical proposals for the use of graphene in electronics cooling. The in-plane thermal conductivity of graphene, 1950 W/mK, is even higher than that of diamond, 895 W/mK, but manufacturing of suffi-

ciently thick graphene layers is presently beyond reach [11].

2.4 Enhanced thermal convection

Convective heat transfer from the chip to the surroundings is significantly smaller than heat transfer to the printed wiring board. The efficiency of convective heat transfer can be increased by the increase of the contact area between the chip and the fluid. The convective heat loss can be enhanced with forced convection. However, this is not a practical solution for handset cooling.

A porous material would provide a tremendous surface area for the convective heat transfer. However, natural convection does not allow the fluid to be forced through a porous material. Instead, folding of the convective surface can be used to increase the effective surface area, and enhance convective heat loss. In the absence of forced convection, folding of the convective surface must not significantly impede the flow of fluid.

Typically, a fin array is used to increase the convective surface area between the heat spreader and the surrounding fluid. Recently, carbon nanotubes have been proposed as possible candidates to produce a microfin structure. Compared to brass or aluminum fin arrays, the advantages of carbon nanotube microfin arrays are lightness, mechanical robustness, and the possibility of integrating fin array manufacturing into current IC processing [12, 13].

Under natural convection, heat dissipation through nanotube microfin arrays has been shown to increase by 11 % [13]. The volume of free air in a handset is, however, small, and it is not clear whether an increase of the contact area between the heat spreader and the surrounding air provides cooling benefits.

2.5 Phase change materials

In contrast to active and passive cooling, PCMs only buffer thermal cycling of the die stack. PCMs protect the die stack against transient power peaks, whereas the steady state of the die stack remains unchanged. However, the requirements for a hypothetical PCM are rather severe: The phase change needs to occur between the steady state operation and the critical operational temperature of the stack. Furthermore, both phases of the PCM should have mechanical properties compatible with the silicon dies and the adhesives in the stack. Moreover, the phase change must require enough energy and be fast enough to significantly affect the temperature of the device [14, 15].

PCMs can be situated in or in the proximity of the die stack. The drawback of in-stack PCMs is that they need to be integrated into the package

assembly process. External PCMs, on the other hand, are less effective due to the separation the of heat sink and the source.

We do not consider PCMs a promising primary temperature management method. However, in microchannel liquid coolers, PCM particle lading or phase change fluids remain a potential option.

2.6 Considered cooling solutions

In the numerical investigation we will consider stack layout and decreased thermal interface resistance. Decreased thermal resistance can be thought of as modeling either advanced adhesive materials, or increased thermal conduction resulting from a well distributed MC flow network bridging the interface.

Chapter 3

Physical model

There are three physical mechanisms underlying heat transfer. When a thermal gradient is present in a solid or in a stationary fluid, heat is conducted. In a moving fluid thermal gradients still cause heat conduction, but thermal energy is also transported together with the medium. Moreover, all surfaces radiate heat, and when the outgoing radiation is less than the incoming radiation, heat is lost from the radiating surface.

3.1 Heat equation

We derive the heat equation starting from the laws of mass and energy conservation [16]. In an arbitrary volume, $\Omega_0 \subset \Omega$, conservation of energy demands

$$\frac{d}{dt}\int_{\Omega_0} dx\, e\rho = -\int_{\partial\Omega_0} dS\, \vec{q}\cdot\vec{n} + \int_{\Omega_0} dx\, \rho f,$$

where e is the density of internal energy, $\rho = \rho(x,T)$ is the density of body, $\vec{q} = \vec{q}(x,T)$ is the heat flux, and $f = f(x)$ is the thermal load density. We apply the divergence theorem on the surface integral term, and move the time derivative inside the integral to obtain

$$\int_{\Omega_0} dx\, \dot{e}\rho = \int_{\Omega_0} dx\, (\rho f - \nabla\cdot\vec{q}),$$

where the material time derivative is denoted by a point. As this holds for an arbitrary volume in Ω, we must have

$$\dot{e}\rho + \nabla\cdot\vec{q} - \rho f = 0 \quad \text{in } \Omega. \tag{3.1}$$

Since the internal energy is dependent on both the pressure, p, and temperature, the time derivative of the internal energy becomes

$$\dot{e} = \frac{\partial e}{\partial T}\dot{T} + \frac{\partial e}{\partial p}\dot{p}.$$

We set $\dot{p} = 0$ and obtain

$$\dot{e} = \left(\frac{\partial e}{\partial T}\right)_p \dot{T} = c_p \dot{T} = c_p \left(\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T\right), \tag{3.2}$$

where $c_p = c_p(x, T)$ is the heat capacity at constant pressure and $\vec{u} = \vec{u}(x, T, t)$ is the velocity of the medium.

Together with the constitutive assumption $\vec{q} = -k \nabla T$, Eqs. (3.1) and (3.2) give

$$\rho c_p \left(\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T\right) - \nabla \cdot (k \nabla T) = \rho f \quad \text{in } \Omega,$$

in the general case. Here $k = k(x, T)$ is the thermal conductivity, and is continuous in both Ω and T.

In the Nokia thermal environment (NVTTE4) convection is modeled by an anisotropic thermal conductivity. For this reason, we assume that $\vec{u} = 0$ from now onwards. The transient heat equation is then

$$\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = \rho f \quad \text{in } \Omega, \tag{3.3}$$

together with suitable boundary conditions.

When $k(x, T)$ has discontinuities, continuity of the solution of Eq. (3.3) and conservation of energy must be explicitly enforced. To do this, we divide Ω into subdomains Ω_i, $i = 1, \ldots, N$ such that

(i)	$\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i,$	
(ii)	$\Omega_i \cap \Omega_j = \emptyset$	$\forall\, i \neq j,$
(iii)	k is continuous in Ω_i	$\forall\, i = 1, \ldots, N.$

We denote by $k_{\bar{\Omega}_i}$ the continuous extension of k to $\bar{\Omega}_i$ with regards to x and T, by $T_{\bar{\Omega}_i}$ the continuous extension of T to $\bar{\Omega}_i$ with regards to x, and by $\nabla T_{\bar{\Omega}_i}$ the continuous extension of ∇T to almost all of $\bar{\Omega}_i$ with regards to x. By $\vec{n}_{\partial\Omega_i}$ we denote the outward surface normal to Ω_i.

Equation (3.3) with discontinuous $k(x, T)$ then becomes

$$\begin{aligned}
\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) &= \rho f && \text{in } \Omega_i \quad \forall\, i = 1, \ldots, N, \\
T_{\bar{\Omega}_i} &= T_{\bar{\Omega}_j} && \text{on } \partial\Omega_i \cap \partial\Omega_j \quad \forall\, i \neq j, \\
k_{\bar{\Omega}_i} \nabla T_{\bar{\Omega}_i} \cdot \vec{n}_{\partial\Omega_i} + k_{\bar{\Omega}_j} \nabla T_{\bar{\Omega}_j} \vec{n}_{\partial\Omega_j} &= 0 && \text{on } \partial\Omega_i \cap \partial\Omega_j \quad \forall\, i \neq j,
\end{aligned} \tag{3.4}$$

together with boundary conditions. The NVTTE4 model includes both isolated boundaries and boundaries with fixed temperature as well as boundaries with mixed boundary conditions. The boundary conditions are

$$\begin{aligned} T &= T_{\text{Ext}} && \text{on } \partial\Omega_D, \\ \nabla T \cdot \vec{n} &= 0 && \text{on } \partial\Omega_N, \\ -k\,\nabla T \cdot \vec{n} &= K\,(T - T_{\text{Ext}}) && \text{on } \partial\Omega_M. \end{aligned} \tag{3.5}$$

Here $\partial\Omega_D$ is the boundary with fixed temperature, $\partial\Omega_N$ is the isolated boundary and $\partial\Omega_M$ is the boundary were the heat flux is dependent on the temperature at the boundary, and $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_M$.

To find the weak formulation of the heat equation at every instant t, we define the linear variety $\mathcal{T}$ and variational space $\mathcal{T}_0$

$$\begin{aligned} \mathcal{T} &= \{T \in H^1(\Omega) \mid \gamma_D(T) = T_{\text{Ext}}\} \\ \mathcal{T}_0 &= \{V \in H^1(\Omega) \mid \gamma_D(V) = 0\}, \end{aligned}$$

and demand that $T \in \mathcal{T}$.

We can now multiply Eq. (3.4) with a test function, $V \in \mathcal{T}_0$, and integrate over Ω to obtain

$$\int_\Omega dx\, \rho c_p \frac{\partial T}{\partial t} V - \int_\Omega dx\, \nabla \cdot (k \nabla T)\, V = \int_\Omega dx\, \rho f V.$$

Partitioning the second term of the left hand side into integrals over the Ω_i:s we obtain

$$\int_\Omega dx\, \rho c_p \frac{\partial T}{\partial t} V - \sum_{i=1}^{N} \int_{\Omega_i} dx\, \nabla \cdot (k \nabla T) V = \int_\Omega dx\, \rho f V.$$

Using Green's formula on the second term on the left hand side it becomes

$$\sum_{i=1}^{N} \int_{\Omega_i} dx\, k \nabla T \cdot \nabla V - \sum_{i=1}^{N} \int_{\partial\Omega_i} dS\, k_{\bar{\Omega}_i} \nabla T_{\bar{\Omega}_i} \cdot V \vec{n}_{\partial\Omega_i}.$$

The surface integral terms can be further partitioned into

$$\sum_{i=1}^{N}\left[\int_{\partial\Omega\cap\partial\Omega_i} dS\, k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n}_{\partial\Omega_i}V + \sum_{j=1,j\neq i}^{N}\int_{\partial\Omega_i\cap\partial\Omega_j} dS\, k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n}_{\partial\Omega_i}V\right] =$$

$$= \sum_{\alpha=\{D,N,M\}}\sum_{i=1}^{N}\int_{\partial\Omega_\alpha\cap\partial\Omega_i} dS\, k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n}_{\partial\Omega_i}V +$$

$$+\sum_{i=1}^{N-1}\sum_{j>i}^{N}\left[\int_{\partial\Omega_i\cap\partial\Omega_j} dS\, k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n}_{\partial\Omega_i}V + \int_{\partial\Omega_i\cap\partial\Omega_j} dS\, k_{\bar{\Omega}_j}\nabla T_{\bar{\Omega}_j}\cdot\vec{n}_{\partial\Omega_j}V\right]$$

$$= \sum_{\alpha=\{D,N,M\}}\int_{\partial\Omega_\alpha} dS\, k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n}_{\partial\Omega_i}V +$$

$$+\sum_{i=1}^{N-1}\sum_{j>i}^{N}\int_{\partial\Omega_i\cap\partial\Omega_j} dS\left(k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i} - k_{\bar{\Omega}_j}\nabla T_{\bar{\Omega}_j}\right)\cdot\vec{n}_{\partial\Omega_i}V,$$

and the final term vanishes, as $k_{\bar{\Omega}_i}\nabla T_{\bar{\Omega}_i}\cdot\vec{n} = k_{\bar{\Omega}_j}\nabla T_{\bar{\Omega}_j}\cdot\vec{n}$ according to Eq. (3.4).

By applying the boundary conditions on $\partial\Omega_D$, $\partial\Omega_N$, and $\partial\Omega_M$ given in Eq. (3.5), we obtain the weak formulation of the nonlinear thermal problem: Find $T\in\mathcal{T}=\{T\in H^1(\Omega)\mid\gamma_D(T)=T_{\text{Ext}}\}$ such that

$$d(T,V)+a(T,V)=\ell(V)\quad\forall\, V\in\mathcal{T}_0=\{V\in H^1(\Omega)\mid\gamma_D(V)=0\},\quad(3.6)$$

where

$$d(T,V)=\int_\Omega dx\,\rho c_p(x,T)\frac{\partial T}{\partial t}V,$$

$$a(T,V)=\sum_{i=1}^{N}\int_{\Omega_i} dx\,k(x,T)\nabla T\cdot\nabla V+\int_{\partial\Omega_M} dS\,KTV,$$

$$\ell(V)=\int_\Omega dx\,\rho fV+\int_{\partial\Omega_M} dS\,KT_{\text{Ext}}V.$$

3.2 Linear elasticity

The displacement field can be calculated from the Navier equations, which describe the behavior of a solid. To derive the Navier equations [17], we consider a point, x, subject to a displacement, u. The displacement moves x to $x'=x+u$. Another point, $x+dx$, in the neighborhood of x, is on the other hand displaced to $x'+dx'$. The derivative of u then becomes $du_j=u_{j,i}dx_i$ using index notation.

The square of the distance between the points before the deformation is $ds^2 = dx_j\, dx_j$. After the displacement the square of the distance is

$$\begin{aligned} ds'^2 = dx_j'^2 &= (dx_j + u_{j,i}dx_i)^2 = \\ &= dx_j^2 + u_{j,i}dx_i dx_j + u_{i,j}dx_i dx_j + u_{k,i}u_{k,j}dx_i dx_j = \\ &= dx_j^2 + 2\epsilon_{ij}dx_i dx_j. \end{aligned}$$

In the above, we have renamed the dummy indices as necessary and defined

$$\epsilon_{ij} = \frac{1}{2}\left(u_{j,i} + u_{i,j} + u_{k,i}u_{k,j}\right).$$

When $u_{m,n} \ll 1$, the term $u_{k,i}u_{k,j}$ is insignificant compared to the linear terms. Hence, we obtain the linear strain tensor ϵ. The linear strain tensor inherits the symmetric character of the nonlinear strain tensor, and is defined as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right).$$

To derive the stresses induced in a body, Ω we consider an arbitrary volume, Ω_0, of Ω. When Ω_0 is at rest, the stress, σ, acting on the boundary of the volume and the body forces, $\vec{f}$, acting on the interior of Ω_0, must cancel. Thus,

$$\int_{\partial\Omega_0} dS\, \sigma\vec{n} + \int_{\Omega_0} dx\vec{f} = 0.$$

Applying the divergence theorem on the first term on the left we get

$$\int_{\Omega_0} dx \left(\nabla\cdot\sigma + \vec{f}\right) = 0.$$

As this holds for any $\Omega_0 \subset \Omega$ we must have

$$-\nabla\cdot\sigma = \vec{f}. \tag{3.7}$$

To solve Eq. (3.7) we also need a relation describing the material behavior. The constitutive relation ties stress and strain together. For small strains, we can use Hooke's law

$$\sigma = \frac{E}{1+\nu}\epsilon + \frac{E\nu}{(1+\nu)(1-2\nu)}\mathrm{tr}(\epsilon)I \tag{3.8}$$

as the constitutive equation. Here E is the modulus of elasticity, and can be obtained by dividing the tensile stress σ by the longitudinal strain ϵ produced by it. ν is the Poisson ratio, the ratio of the lateral contraction to the longitudinal strain under pure tension.

3.3 Thermal strain

When heated from T_0 to T, an unconstrained isotropic body, Ω, will expand [17]. The linear approximation of the thermal strains induced in the body is

$$\epsilon^T = \alpha \Delta T I. \tag{3.9}$$

We use the superposition principle to determine the strain produced by both stress and temperature increase,

$$\epsilon^F = \epsilon + \epsilon^T, \tag{3.10}$$

where ϵ is the strain produced by the stress σ and ϵ^F is the final strain.

Substituting the superposition principle, Eq. (3.10), and the thermal strain, Eq. (3.9), into Hooke's law, Eq. (3.8), we obtain

$$\begin{aligned}
\sigma^F &= \frac{E}{1+\nu}\epsilon^F + \frac{E\nu}{(1+\nu)(1-2\nu)}\mathrm{tr}(\epsilon^F)I \\
&= \sigma + \frac{E}{1+\nu}\alpha\Delta T I + \frac{E\nu}{(1+\nu)(1-2\nu)}\alpha\Delta T \mathrm{tr}(I) I \\
&= \sigma + \left[\frac{1}{1+\nu} + \frac{3\nu}{(1+\nu)(1-2\nu)}\right]\alpha\Delta T E I \\
&= \sigma + \frac{E\alpha\Delta T}{1-2\nu} I.
\end{aligned}$$

By further expressing the internal energy as the integral over generalized force and displacement, we obtain

$$\begin{aligned}
W &= \int_0^{\epsilon} \sigma^F : d\epsilon^F \\
&= \int_0^1 \left[t\left(\frac{E}{1+\nu}\epsilon + \frac{E\nu}{(1+\nu)(1-2\nu)}\mathrm{tr}\left(\epsilon\right) I\right) - \frac{E\alpha\Delta T}{1-2\nu}\right] : \epsilon dt \\
&= \frac{1}{2}\left(\frac{E}{1+\nu}\epsilon : \epsilon + \frac{E\nu}{(1+\nu)(1-2\nu)}\mathrm{tr}\left(\epsilon\right)^2\right) - \frac{E\alpha\Delta T}{1-2\nu}\mathrm{tr}\left(\epsilon\right).
\end{aligned}$$

The work done by the system is

$$\int_\Omega dx W_{\mathrm{Ext}}\left(u\right) = \int_\Omega dx f_i u_i + \int_{\partial\Omega} dS s_i u_i,$$

where f is the body force and s is the surface force. The total energy of the system is

$$J\left(u\right) = \int_\Omega dx W\left(u\right) - \int_\Omega dx W_{\mathrm{Ext}}\left(u\right).$$

From here we can identify the bilinear part $b(\cdot\,,\cdot\,)$ and the load $g(\cdot\,)$,

$$b(u,v) = \int_\Omega dx \sigma^F : \epsilon,$$
$$g(u) = \int_\Omega dx \frac{E\alpha\Delta T}{1-2\nu}\mathrm{tr}\,(\epsilon) + \int_\Omega dx f_i u_i + \int_{\partial\Omega} dS s_i u_i.$$

The weak formulation of the problem is then:
Find $u \in \mathcal{U} = \{u \in \left[H^1(\Omega)\right]^3 \mid \gamma_D(u) = 0\}$ such that

$$b(u,v) = g(v) \tag{3.11}$$

for all $v \in \mathcal{U}$.

When solving the thermal strain problem we must first solve the temperature distribution, Eq. (3.6), and use the obtained solution to calculate the thermal strain.

Chapter 4

Numerical solution

In mathematical physics and engineering, several problems are formulated as partial differential equations such as Eq. (3.1) above. While the problem is infinite dimensional, numerical methods permit only a finite dimensional approximation to be computed. The finite element method is one scheme to obtain a finite dimensional approximation of the original equation.

A finite element approximation of a nonlinear problem inherits the nonlinearity of the initial problem. The resulting nonlinear system of equations is then solved as a sequence of linear matrix equations. The solving of nonlinear equations is therefore reduced to solving of linear systems.

Solution methods for linear systems of equations can be divided into two general classes: direct and iterative methods. A fundamental difference is that a direct method solves the problem in one step, whereas an iterative method requires several steps to converge.

4.1 The finite element method

The Ritz-Galerkin method provides a finite dimensional approximation from the weak formulation of the problem by restricting the solution to some appropriate finite dimensional subspace [18].

In the finite element method, we constrain the solution to a piecewise polynomial function. In practice, the finite element space is determined by the choice of discretization of the computational domain into elements. This results in a method that is comparatively easy to implement, while inner product evaluations of the basis functions is comparatively inexpensive.

Evaluation of the inner product is done by considering the contribution of each element separately, and assembling the contribution into the system matrix. Shape functions are not integrated over the local element, instead an affine transformation is used to map the local basis functions to a reference element. The integral is then evaluated on the reference element, and the affine mapping can be used to obtain the value of the inner product over the

local element.

To obtain a linear problem from Eq. (3.6), we introduce the linearization $\tilde{a}$ of a such that $k(x,T) = k(x,\hat{T})$, and $\tilde{d}$ of d such that $c_p(x,T) = c_p(x,\hat{T})$, and $\rho(x,T) = \rho(x,\hat{T})$ for a given $\hat{T} = \hat{T}(x,t)$. Usually $\hat{T}$ is the solution of the previous step of the nonlinear iteration.

Here, we use a piecewise linear finite element space. We associate with each node, n_k, in a tetrahedrization, $\mathcal{C}_h$, of Ω a hat function such that $\psi_j(n_k) = \delta_{jk}$ and for each tetrahedron K, $\psi_j|_K \in P_1(K)\ \forall K \in \mathcal{C}_h$. Here $P_1(K)$ denotes the linear functions on K. We also order the nodes such that the interior nodes are indexed $1,\ldots,m$ and the boundary nodes are indexed $m+1,\ldots,N$.

Now, $T_h \in \mathcal{T}_h = \{V \in \mathcal{T} \mid V|_K \in P_1(K)\ \forall K\}$ is a solution in $\mathcal{T}_h$ provided that

$$\tilde{d}(T_h, V) + \tilde{a}(T_h, V) = \ell(V) \qquad \forall\, V \in \mathcal{T}_{h0}, \tag{4.1}$$

with $\gamma_D(T_h) = T_{\text{Ext}}$, where T_{Ext} is a given temperature on $\partial\Omega_D$ and is the trace of some function in $\mathcal{T}_h$. $T(x,0) \in \mathcal{T}_h$ is given and $\mathcal{T}_{h0} = \{V \in \mathcal{T}_0 \mid V|_K \in P_1(K)\ \forall K\}$.

In the finite element basis, $\{\psi_1,\ldots,\psi_N\}$, Eq. (4.1) is equivalent to

$$\begin{aligned}
&\tilde{d}(\sum_{j=1}^{N} z_j\psi_j, V) + \tilde{a}(\sum_{j=1}^{N} z_j\psi_j, \psi_i) = \\
&= \sum_{j=1}^{N} \dot{z}_j \tilde{s}(\psi_j, V) + \sum_{j=1}^{N} z_j \tilde{a}(\psi_j, \psi_i) = F(\psi_i) \qquad \forall\, i \in 1,\ldots,m,
\end{aligned} \tag{4.2}$$

where $\gamma_D(\sum_{j=1}^{N} z_j\psi_j) = T_{\text{Ext}}$ and $T_h = \sum_{j=1}^{N} z_j\psi_j$, with $z_j = z_j(t)$ to account for the time evolution, and

$$\tilde{s}(\psi_j, \psi_i) = \int_\Omega dx\, \rho(x,\hat{T}) c_p(x,\hat{T}) \psi_j \psi_i.$$

We collect the interior degrees of freedom into the vector $\vec{z} \in \Re^m$, and the boundary values into $\vec{y} \in \Re^{(N-m)}$, such that

$$\begin{aligned}
\vec{z}_i &= z_i, && \forall\, i = 1,\ldots,m, \text{ and} \\
\vec{y}_i &= z_{i+m}, && \forall\, i = 1,\ldots,N-m.
\end{aligned}$$

We can then write Eq. (4.2) as the matrix equation

$$\begin{bmatrix} M & S & A & B \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \dot{\vec{z}} \\ \dot{\vec{y}} \\ \vec{z} \\ \vec{y} \end{bmatrix} = \begin{bmatrix} \vec{f} \\ \dot{\vec{g}} \\ \vec{g} \end{bmatrix}, \tag{4.3}$$

together with the initial condition $\vec{z}_i(0) = T(n_i, 0)$. Here $M \in \Re^{m\times m}$ is the mass matrix, $S \in \Re^{m\times(N-m)}$ is the mass matrix corresponding to the

boundary nodes, $A \in \Re^{m\times m}$ is the stiffness matrix, $B \in \Re^{m\times(N-m)}$ is the stiffness matrix corresponding to the boundary nodes, $\vec{f} \in \Re^m$ is the load, and $\vec{g} \in \Re^{(N-m)}$ is the boundary condition. These are given by

$$\begin{aligned} M_{ij} &= \tilde{s}(\psi_j, \psi_i), \\ S_{ij} &= \tilde{s}(\psi_{m+j}, \psi_i), \\ A_{ij} &= \tilde{a}(\psi_j, \psi_i), \\ B_{ij} &= \tilde{a}(\psi_{m+j}, \psi_i), \\ \vec{f}_i &= \ell(\psi_i), \\ \vec{g}_i &= T_{\text{Ext}}(n_{m+i}), \\ \vec{z}_i &= T_h(n_i), \text{ and} \\ \vec{y}_i &= T_h(n_{m+i}). \end{aligned}$$

It is clear that $\vec{y} = \vec{g}$ and $\dot{\vec{y}} = \dot{\vec{g}}$. Equation (4.3) then becomes

$$M\dot{\vec{z}} + A\vec{z} = \vec{b}, \tag{4.4}$$

where $\vec{b} = \vec{f} - S\dot{\vec{g}} - B\vec{g}$. For the steady state case, $\dot{\vec{z}} = \dot{\vec{g}} = \vec{0}$ and Eq. (4.4) becomes

$$A\vec{z} = \vec{b}. \tag{4.5}$$

4.2 Time discretization

Both the initial temperature distribution and the boundary conditions must be known to obtain the transient behavior of the heat equation. The time interval to be considered is then divided into time steps, Δt and some scheme is used to approximate the new temperature distribution.

One such choice is the backward difference formula (BDF). The BDF methods are implicit and stable up to sixth order. The accuracy of the BDF method increases with increasing order [19, 20].

To derive the second order BDF for Eq. (4.4) we set $t_k = k\,\Delta t$, where $k = 1, 2, \ldots, N$ labels the time steps. We then expand $\vec{z}(t)$ to second order around t_{k+1},

$$\vec{z}(t) = \vec{z}(t_{k+1}) + \dot{\vec{z}}(t)(t - t_{k+1}) + \frac{1}{2}\ddot{\vec{z}}(t)(t - t_{k+1})^2.$$

We fix $t = t_k$ and denote $\vec{z}^k = \vec{z}(t_k)$. Expanding the second order time derivative recursively with the backward difference $\Delta t\,\dot{\vec{z}}^{k+1} = \vec{z}^{k+1} - \vec{z}^k$ we obtain

$$\dot{\vec{z}}^{k+1} = \frac{1}{\Delta t}\left(\frac{3}{2}\vec{z}^{k+1} - 2\vec{z}^k + \frac{1}{2}\vec{z}^{k-1}\right). \tag{4.6}$$

We set $M = M^{k+1}$, $A = A^{k+1}$, and $\vec{b} = \vec{b}^{k+1}$ in Eq. (4.4). Combining this with (4.6) we get

$$\left(\frac{1}{\Delta t}M^{k+1} + \frac{2}{3}A^{k+1}\right)\vec{z}^{k+1} = \frac{2}{3}\vec{b}^{k+1} + \frac{1}{\Delta t}M^{k+1}\left(\frac{4}{3}\vec{z}^{k} - \frac{1}{3}\vec{z}^{k-1}\right). \quad (4.7)$$

4.3 Conjugate gradients

The conjugate gradient (CG) iteration solves symmetric positive definite systems of equations with well separated clusters of eigenvalues extremely fast. Several similar methods have been developed since the introduction of CG, in order to iteratively solve systems lacking these properties [21].

Let $A \in \Re^{m\times m}$ be a real, nonsingular, and symmetric. We wish to solve

$$A\vec{z} = \vec{b}, \quad (4.8)$$

with the solution $\vec{z} = A^{-1}\vec{b}$ using an iterative method. For this purpose, we define the error at step n as $\vec{e}_n = \vec{z} - \vec{z}_n$. We further define the n:th Krylov subspace, $\mathcal{K}_n$, generated by $\vec{b}$ as

$$\mathcal{K}_n = \left\langle \vec{b}, A\vec{b}, \ldots, A^{n-1}\vec{b} \right\rangle.$$

When A is a positive definite symmetric matrix, the A-norm is defined by

$$\|\vec{z}\|_A = \sqrt{\vec{z}^T A\vec{z}}.$$

The CG method then generates the unique sequence of vectors, $\vec{z}_n \in \mathcal{K}_n$, that $\|\vec{e}_n\|_A$ is minimized at every step. Moreover, the residuals are orthogonal, and the search directions, $\vec{p}$ are A-conjugate

$$\vec{p}_n^T A\vec{p}_j = 0 \qquad (j < n).$$

A modification to the CG method is the biconjugate gradient method (BCG). In contrast to the CG method, BCG can solve Eq. (4.8) for asymmetric A. This property is obtained by not minimizing $\|\vec{e}_n\|_A$ at every step, instead, for $\vec{z}_n \in \mathcal{K}_n$ we require that

$$\vec{e}_n \perp \left\langle \vec{w}, A^T\vec{w}, \ldots, (A^T)^{n-1}\vec{w} \right\rangle,$$

where $\vec{w}$ satisfies $\vec{w}^T\vec{b} = 1$.

A further improvement over the BCG method is BiCGSTAB, introduced by van der Vorst in 1992 [22]. BiCGSTAB has a significantly smother convergence than BCG.

4.4 Preconditioning

The convergence of the matrix iteration depends on the condition number of the matrix. In a well conditioned case, convergence is fast, while convergence can be slow or absent for poorly conditioned systems. Preconditioning can be used to transform a poorly conditioned system into a better behaved one. The condition number of a matrix is defined as $\kappa(A) = \|A\| \, \|A^{-1}\|$.

The solution of $A\vec{z} = \vec{b}$ remains unchanged when the system is multiplied from the left by a matrix M^{-1}, where $M \in \Re^{m \times m}$ and nonsingular. The new system,

$$M^{-1}A\vec{z} = M^{-1}\vec{b},$$

depends on the conditioning properties of $M^{-1}A$ instead of those of A.

The proper choice of the preconditioning matrix M^{-1} is more of an art than a science. The useful preconditioners are close enough to A^{-1} so that the condition number of $M^{-1}A$ is well behaved, while M is still easy to invert.

The iterative solvers require several matrix-vector multiplications. In order for the iteration to be effective, this operation must be inexpensive. A successful preconditioner must therefore be sparse, or have other exploitable structure.

Preconditioners based on incomplete LU-factorization (ILU) are an often used class of preconditioners. In general, LU-factorization results in full lower and upper triangular matrices. To obtain sparse preconditioners we must somehow decide how to apply a sparsity pattern on the preconditioner.

One strategy is to decide a fill level, p, for the ILU factorization. The zeroth level ILU preconditioner, ILU(0) is then constructed by Gaussian elimination, and nonzero elements are only accepted in positions where A has nonzero elements. For the ILU(p) preconditioner, we assign every matrix entry an fill level l_{ij}, and ignore nonzero elements with fill level above p [23]. Initially, the fill level of every matrix entry is set as

$$\begin{aligned} l_{ij} &= 0 && \text{if } A_{ij} \neq 0, \\ l_{ij} &= \infty && \text{else.} \end{aligned}$$

As the Gaussian elimination progresses, the fill level of the matrix entry is updated along with the actual matrix entry by the formula

$$l_{ij} = \min\{l_{ij}, l_{ik} + l_{kj} + 1\}.$$

An ILUT preconditioner, on the other hand, is constructed by only admitting new nonzero matrix elements if the element would exceed a previously specified threshold. This approach typically results in a preconditioner that more closely resembles the actual problem at hand. In contrast with the ILU(p) preconditioner, however, the memory required for storage of an ILUT preconditioner cannot be determined in advance.

Chapter 5

Computational Model

In this chapter we present the simulated stack layouts, the simulation environment, and the tools used for simulation.

5.1 Stack layouts

In this section we investigate the simulated stack layouts. We begin by investigating three different chip stacks (case 1, 2, and 3), with four variants (A, B, C, and D) each. We then further investigate case 3 as it provides the most demanding case for thermal management. Finally, we simulate a stack designed based on results from the earlier simulations.

A chip stack consists of a number of chips mounted on top of a substrate, as shown in Fig. 5.1. A typical stacked chip is a silicon plate with edge length $4 - 9$ mm and a thickness of $50 - 200$ µm. Between the silicon chips there is a layer of adhesive, typically $3 - 15$ µm thick. In this work, we assume an adhesive thickness of 5 µm. Unless otherwise stated, the chips are stacked with their midpoints on top of each other.

Power dissipation is assumed to be uniform in a $1 \times 1 \times 0.05$ mm^3 region of the chip, typically located at the center of the upper surface of the chip. The power footprints of different chips is presented in Fig. 5.2, and the total power dissipated per chip is presented in Tab. 5.1.

The chips are stacked on a silicon plate, the substrate. The entire bottom surface of the substrate is covered with a matrix of solder balls. The solder ball pitch is 1.0 mm. In most variations we assume that the volume under the substrate has an epoxy underfill, but we also investigate one variant (variation C, cf. Tab. 5.3) where the underfill volume is replaced with air.

The three different cases are presented in Fig. 5.3. The first case consists of a memory chip stack. The memory chip stack includes two active chips mounted on top of substrate. In addition, an interposer chip has been inserted between the active chips. Although the processing engine of the memory chip stack does not include a N&S Bridge, the corresponding power

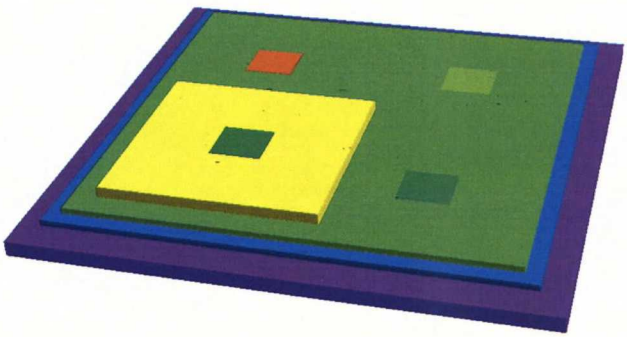

Figure 5.1: Chip stack corresponding to case 2. Note in particular the power dissipating regions in the chips and the off-center modem chip. The encapsulant surrounding the modem chip is omitted for clarity.

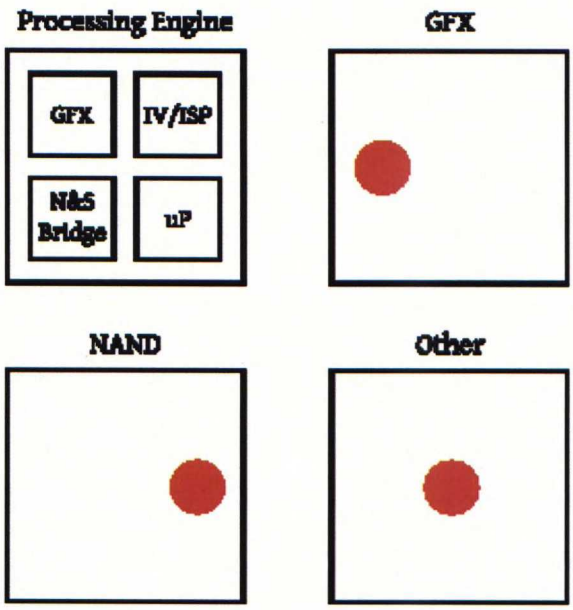

Figure 5.2: Schematic of the assembly of the processing engine. Hotspots of different chips is also shown.

Table 5.1: Chip power dissipation. All chips are not present in every chip stack.

Chip	P [W]
DRAM0 - DRAM3	0.20
Modem	0.32
GFX	0.20
IV/ISP	0.26
uP	0.18
N- & S- Bridge	0.18
NAND	0.06

is dissipated in internal connectivity and I/O. Note in particular that the DRAM module consists of four DRAM chips and has a total power dissipation of 0.80 W, and that the processing engine similarly has a total power dissipation of 0.82 W.

In case 2, an off-center modem chip is added on top of the DRAM module. The modem chip is surrounded by an encapsulant. Power dissipation in the modem chip is an additional 0.32 W. The total power dissipated in the modem chip stack is 1.94 W.

Case 3 consists of a complete integrated stack, and is presented in Fig. 5.3. In contrast to case 1 and 2, the includes four individual DRAM chips instead of a four DRAM module. The processing engine present in case 1 and 2 is also divided into four separate chips. Power dissipation is detailed in Fig. 5.2 and Tab. 5.1. Total power dissipation is 2 W.

Chip and substrate size for different cases and variants is presented in Tab. 5.2. The differences between the four different variants of each cases 1–3 is presented in Tab. 5.3

In the second phase of the investigation, we simulate the effect of several variables on case 3. Changes in stack design are assembly of the stack upside down, pyramidal design of the stack, and dispersion of the heating regions. The effect of the variables are investigated by a $2^6 \times 3$ factorial design. The different factors are listed below.

Stack describes the vertical ordering of the die stack. It can take the value 0 corresponding to an inversely ordered stack, and 1 corresponding to the standard ordering of case 3 presented in Fig. 5.3.

Pyramid describes the width of the chip stack. Value 0 corresponds to a same size die stack with die surfaces 7×7 mm^2. The high value, 1, corresponds to varying die size. When the die size varies, DRAM surface area is 4.3×4.3 mm^2, Modem, GFX, IV/ISP, uP, and North- & South Bridge surface area is 6×6 mm^2. NAND die size is 7×7 mm^2

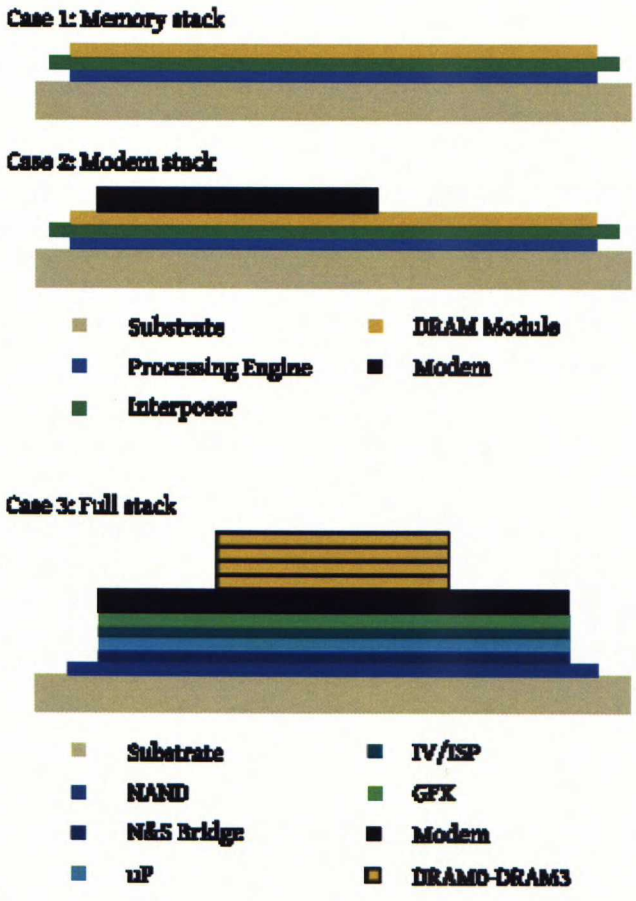

Figure 5.3: Schematic of stack configuration for different cases.

Table 5.2: Chip and substrate size.

Chip	Volume [mm^3]
Substrate (case 1, 2, 3A)	$10 \times 10 \times 0.3$
Substrate (case 3B, 3C, 3D)	$9 \times 9 \times 0.3$
Interposer (case 1, 2)	$9 \times 9 \times 0.1$
Processing Engine (case 1, 2)	$8.7 \times 8.7 \times 0.1$
DRAM module (case 1, 2)	$8.4 \times 8.4 \times 0.1$
DRAM0 - DRAM3 (case 3)	$9 \times 9 \times 0.1$
Modem (case 2)	$4.2 \times 4.2 \times 0.2$
Modem (case 3)	$6 \times 6 \times 0.2$
GFX, IV/ISP, uP, N&S Bridge (case 3)	$6 \times 6 \times 0.1$
NAND (case 3)	$7 \times 7 \times 0.1$

Table 5.3: Summary of differences between variations. z_{adhesive} is the adhesive thickness, underfill describes the underfill material used in the simulation and the different options for heat spreader (HSP) size are defined in Tab. 5.4. $x_{\text{substrate}}$ is the edge length of the substrate edge.

Variation	z_{adhesive} [µm]	Underfill	HSP size	$x_{\text{substrate}}$ [mm] [a)]
A	5	epoxy	HSP 2	10
B	10	epoxy	HSP 2	9
C	10	air	HSP 2	9
D	10	epoxy	HSP 1	9

[a)] Substrate size is only varied for case 3.

regardless of the value of the variable.

Aligned takes value 0 or 1. The low value corresponds to in-plane distribution of the hotspots, while all hotspots are centered for the high value.

TIM describes the thermal conductivity of the thermal interface material layer between die stack and heat spreader. The permissible values are, 1.0 W/mK and 2.0 W/mK.

Adhesive describes the thermal conductivity of the adhesive layer separating dies in the stack. It takes two values, 1.0 W/mK and 2.0 W/mK.

Underfill describes the thermal conductivity of the underfill material. In contrast with TIM and Adhesive. Underfill thermal conductivity takes three values: that of still air, 0.025 W/mK, and adhesives, 1.0 W/mK and 2.0 W/mK.

Chimney Describes the heat dissipation through the heat spreader. The low value corresponds to the NVTTE4 and the high value presents an increase of 14 %. Increased thermal dissipation is modeled by increasing the perpendicular thermal conductivity of the chimney above the heat spreader from 0.035 W/mK to 0.04 W/mK.

5.2 Environment

In order to set up a descriptive model of the chip stack, we need to know the properties of both the environment and the stack.

The simulation environment cannot be unlimited in size. We must instead provide an abstract representation of the external environment at the boundary of the simulation. Determining where and how this simplification is done has a significant effect on both the accuracy of the results and the amount of computational resources required. Choosing a suitable boundary condition requires familiarity with both the physical problem at hand and its mathematical formulation.

In this investigation the NVTTE4 is used. The NVTTE4 consists of a printed wiring board (PWB) with a volume of air on top of it. The chip stack is placed at the mid point of the PWB. A layer of thermal interface material (TIM) connected to a heat spreader (HSP) is added on top of the chip stack. Figure 5.4 is a schematic presentation of the NVTTE4. Size and material parameters of the environment are presented in Tab. 5.4. The sides of the PWB and the top block are assumed to be adiabatic. The top is assumed to be at a fixed ambient temperature, $T_{\text{Ext}} = 60$ °C. The bottom boundary condition is dissipating, with heat transfer coefficient $K = 10$ W/m^{2}K and $T_{\text{Ext}} = 60$ °C.

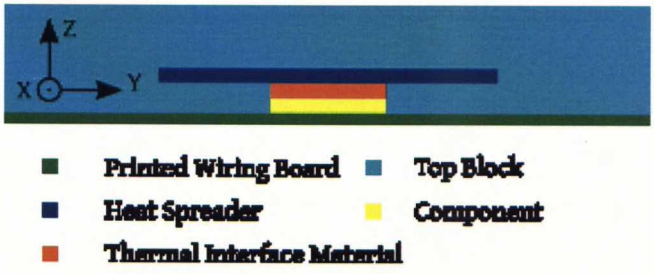

Figure 5.4: Schematic of NVTTE4, see Tab. 5.4 for details.

Table 5.4: Characteristics of the thermal simulation environment.

Region	x [mm]	y [mm]	z [mm]	k_{xx} [W/mK]	k_{zz} [W/mK]
PWB	102	76	1	60	1
Top block	102	76	5	0.015	0.035
TIM	a)	a)	0.5	1	1
HSP 1	50 b)	50 b)	2	100	100
HSP 2	c)	c)	2	100	100

a) x and y dimension equal top stack casing.
b) in the initial investigation HSP edge length equals 30 mm
c) x and y dimension equal top stack casing + 10 mm.

The size of the simulation environment is on the order of 100 mm, while typical inter-chip adhesive layers are of the order 10 μm. The difference in scale, 10^4, requires us to consider homogenization of the material.

Taking into account that the thermal resistance of a 10 μm layer of adhesive is comparable to the thermal resistance of a 100 μm silicon chip, adhesive layers cannot generally be ignored. The effective thermal conductivity normal to the stacked adhesive chip layers is

$$k_{\text{Eff}} = \frac{k_1 k_2 (z_1 + z_2)}{k_1 z_2 + k_2 z_1},$$

where k_i is the thermal conductivity and z_i the thickness of layer i. It is also possible to consider the adhesive layers as thermal contact resistances between adjacent silicon domains. In the earlier exploratory stages of the simulation we use homogenization of the material. In the last stage, where we calculate the thermally induced strains in a sample stack, we instead make use of a structured mesh involving thin hexahedral elements to model the interface layers.

Homogenization of the silicon chip and the adhesive layer lower the average temperature of the homogenized region. The maximum temperature of the chip does, however, remain unchanged.

In the NVTTE4, convection and radiation is taken into account by the introduction of a non-isotropic thermal conductivity for air. This simplification leads to the conduction-only-model presented in Eq. (3.4).

Symmetric horizontal integration of chips with similar power footprints results in small temperature gradients over the adhesive layers, which separate the chips. Therefore, the adhesive layers inside the DRAM module and processing engine can be ignored in case 1 and case 2.

Since heat transfer through the sides of the stacked chip package is relatively small, encapsulation of the sides of the chip stack does not significantly affect the package temperature. Encapsulation of the top surface is masked by the thermal resistance of the 500 μm thick TIM layer between the component and the heat spreader. Additional top surface encapsulation of the chip stack is included in the calculations for case 2.

Instead of using round solder balls, we use cubes with an edge length corresponding to the distance between the substrate and the PWB.

The literature concerning the precise boundary conditions for practical electronic thermal characterization is scarce. A typical choice is to assume linear temperature dependence, and use 0 °C boundary conditions. Forced air convection is also commonly assumed, see *e.g.* [24–27]. These assumptions are not appropriate for the enclosed environment of a handset, and the NVTTE4 provides a best guess estimate for the thermal behavior of the environment.

Thermal resistance of interface layers is not explicitly present in the

model. It is instead implicitly included in the total effective perpendicular thermal conductivity of the silicon dies.

Selected material parameters are presented in Tab. 5.5. In addition, the thermal properties of silicon are

$$k = 117.5 - 0.42 \times (\mathrm{T} - 100)\ \mathrm{W/mK},$$
$$c_p = 836 + 0.121 \times (\mathrm{T} + 273) - 1.396 \times 10^7 / (\mathrm{T} + 273)^2\ \mathrm{J/kgK}, \text{ and}$$
$$\rho = 2330\ \mathrm{kg/m^3}.$$

Table 5.5: Thermal properties overview [15, 28–30].

Material	k [W/mK]	c_p [J/kgK]	ρ [kg/m^3]
epoxy & TIM	1	300	2450
solder	36	137	11200
PWB	a)	880	1800
HSP	100	770	3260
air	a)	1009	1.067

a) See Tab. 5.4 for details.

5.3 Simulation tools

5.3.1 Elmer

Elmer is an open source simulation software for multiphysics problems, that is developed by CSC, the Finnish IT Center for Science [31]. The partial differential equations (PDE) describing the physical models are solved by Elmer using the finite element method (FEM).

In FEM, the unknown solution to the PDE is approximated by another function with a finite number of degrees of freedom. The entire geometry of the problem is divided into parts called elements, and the matrix equation to be solved is assembled by considering separate elements individually.

Initially, the temperature dependent material parameters of silicon were implemented in the MATC language included with Elmer. The replacement of the MATC statements with subroutines written in Fortran cut simulation times by half. Body forces are still implemented in MATC to increase readability. As the number of elements concerned is relatively small, this should not significantly affect simulation times.

For the output of reference points and requested information, a new solver is provided. To use the new solver the Elmer solver input file must contain the solver declaration:

```
Solver n
    Equation = String "Reference"
    Procedure = "Reference" "ReferenceValues"
    Filename = String "filename.dat"
    Scalar Field = String "Temperature"
End
```

The solver declaration can include the optional keyword

```
Calculate All [Logical]
```

Solver **Reference** calculates the maximum, minimum, and average values for bodies and boundary conditions. The following optional keywords are provided for bodies and boundary conditions:

```
Reference Name [String]
Reference Calculate [Logical]
```

Reference Name and **Reference Calculate** provide better control of the output data.

The default behavior of the **Reference** solver is to consider only those regions defining the **Reference Name** keyword. This can be overridden by the solver keyword **Calculate All**, which in turn is overridden by the region specific **Reference Calculate** keyword. If **Reference Name** is not provided, the solver by default uses the **Name** keyword. If neither **Reference Name** nor **Name** is provided for a region, it is named **body n** or **bc n**.

Calculation of diffusive heat flux is done with the solver **SaveScalars** included in the release of Elmer. The implementation typically underestimates heat flux by $5 - 10$ %. The underestimation originates from the inherent inaccuracy of estimating the normal derivative of a scalar field in FEM.

When changing the thickness of adhesive or silicon chips in a homogenized silicon chip, the file **ChipstackStuff.f90** must also be edited and recompiled. It is only necessary to edit the subroutines **CondThinLumpedSilicon** and **CondThickLumpedSilicon**, as these describe the thermal conductivity of the adhesive and the thickness of both the silicon and the adhesive layer.

5.3.2 Gmsh

Gmsh is a program with both a graphical and a script based interface for description and meshing of geometries. We have chosen to use Gmsh to describe the chip stack and the NVTTE4 as Gmsh provides a text based interface for description of the repetitive problem geometry [32]. In particular, we can use named variables to relatively easily make changes to the geometry of the simulation. Two different 3D meshing algorithms, Netgen [33] and Tetgen [34] are also available in Gmsh.

The Gmsh description of a geometry is divided into three files. The geometry description file **functions.geo**, common to all cases define re-

peatedly used functions. Each case also includes two additional files, one describing the NVTTE4 environment and another describing the actual chip stack. While the file `stack.geo` is completely case dependent, case dependent statements in `main_stack.geo` describing the NVTTE4 are limited to the `Include` statement needed to load `stack.geo` as well as the definitions of the physical regions. A wire frame geometry of the file `modem.geo` is presented in Fig. 5.5. The file `modem.geo` corresponds to the file `stack.geo` for case 2.

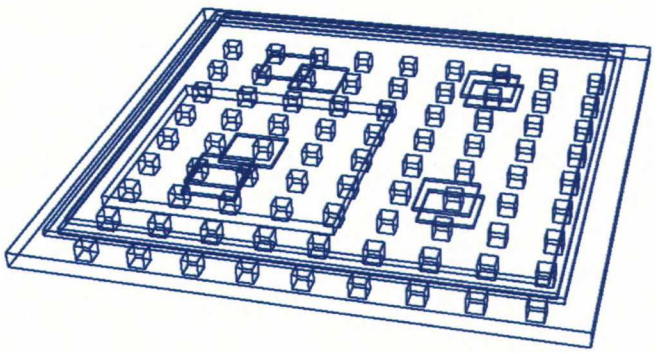

Figure 5.5: Wire frame showing detail of the modem geometry. The underfill volume and the encapsulant are left out.

Due to limited support for control structures in the Gmsh description language, the geometrical description of the problem must be done carefully. Otherwise structures, which are impossible to mesh using a boundary first algorithm, could be introduced. Of the two 3D meshing algorithms provided in Gmsh, Netgen is both faster and produces a higher quality mesh for this problem. It is still recommended to enable additional optimization of the mesh. Details of a mesh generated with Netgen for case 2 is shown in Fig. 5.6. Typically the meshes consist of 1.2-1.5 million first order tetrahedral elements. With a 2.4 GHz processor, mesh generation and optimization takes half an hour. Less than 512 MB memory is sufficient.

The mesh generating algorithms are unfortunately not particularly robust. It is occasionally necessary to change some parameters to enable mesh generation. Repeated manual intervention in mesh generation can become rather time consuming.

Mesh generation was made by Gmsh using the Netgen algorithm and subsequent mesh quality optimization for Γ, where

$$\Gamma \propto \frac{V_{\text{Element}}}{E_{\max} \sum A_{\text{Face}}}.$$

Here V_{Element} is the element volume $E_{\max}$ is the maximum edge length and A_{Face} is the face surface. Γ provides a measure of the quality of the element,

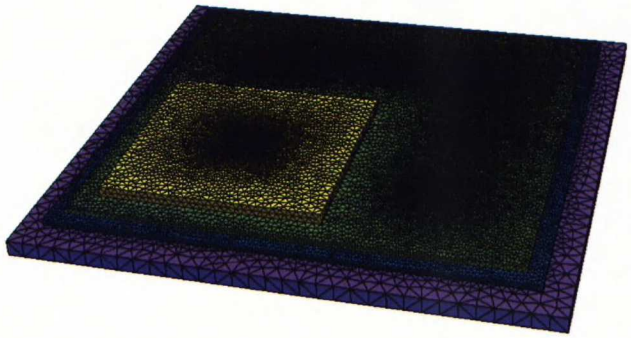

Figure 5.6: Detail of mesh for case 2.

and can be used to spot low quality elements. Tetrahedral elements that have been flattened, slivers give a low value for Γ when calculated. Figures 5.7 and 5.8 presents the mesh quality for a typical mesh. For the worst meshes the lowest quality elements have $\Gamma > 0.01$. The lowest quality elements are rare and unclustered. Conversion from the Gmsh mesh format to the Elmer mesh format can be done with `ElmerGrid`. The Metis graph partitioning programs included in `ElmerGrid` has been used to partition the mesh [35].

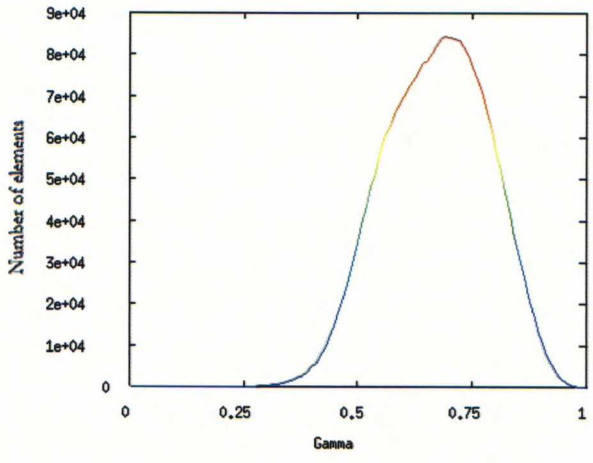

Figure 5.7: Mesh quality for typical tetrahedral mesh.

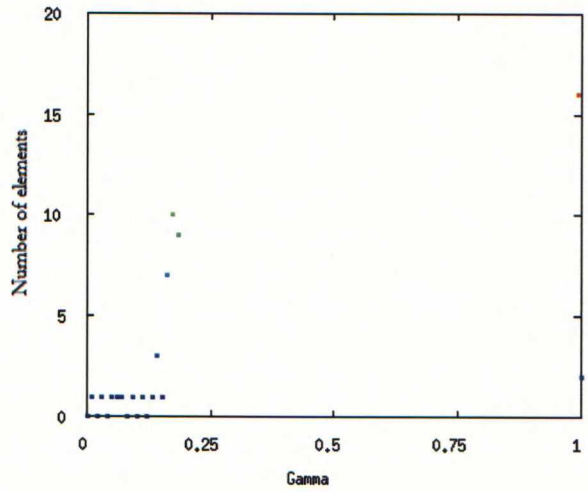

Figure 5.8: Zoom in of distribution of low quality elements from Fig. 5.7.

5.4 Factorial design

Factorial design provides a systematic method to judge the effect of several variables, or factors [36]. In Fig. 5.9 a 2×2 factorial experiment is presented. Main effects and two-factor interaction effects in a multi-factor experiment are interpreted identically as for a 2×2 experiment.

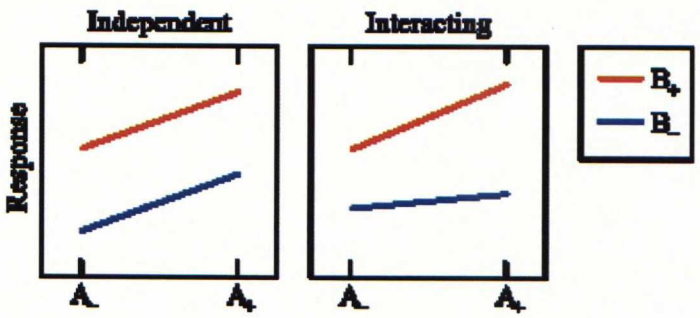

Figure 5.9: Independent and interacting 2×2 factorial design. In the interacting case, an changing factor A from A_- to A_+ gives a larger response when factor $B = B_+$ than when $B = B_-$.

The main effect of a factor is simply the average effect of the increase of a factor. While the average can be calculated over several other varying factors, the combination of values for factors should be compatible.

To present two-factor interactions, averages for the possible combinations of low and high variables are calculated. The result in then plotted as in Fig. 5.9. In the independent case, the response of increasing A from A_- to A_+ is independent of the level of B. The lines corresponding to B_- and B_+ are consequently parallel. Conversely, in the interacting case, the increase of A leads to a larger response if $B = B_+$ than if $B = B_-$, and the line corresponding to B_+ has a larger derivative. The main effect of a factor does not accurately reflect the effect of the factor when interaction effects are present.

A script was used to change the values of the factorial variables in the solver input file, and then to record the maximum temperature of both the stack and the DRAM units.

Chapter 6

Results

In this chapter we present central results calculated for this work. We first present results from a preliminary investigation of three chip stacks. We then present more in detail results for the third chip stack from the preliminary phase. Finally, we calculate the thermally induced stress in a stack design based on the guidelines from the earlier phases.

6.1 Temperature distribution

Figure 6.1 presents the temperature distribution of case 3, variation B. The temperature distribution of the chip stack is shown in Fig. 6.2. Figure 6.3 presents the time evolution of the component maximum temperature for variation A. The maximum temperature follows a $1 - e^{-ct}$ time dependence when heated from a uniform temperature distribution corresponding to the ambient temperature.

The reference points and requested information for cases 1, 2 and 3 are presented in tables A.1 through A.4. The reference points are defined as follows:

T_{J} is the maximum temperature of the die, known as junction temperature.

T_{B} is the maximum temperature of the printed wiring board 1 mm away from the component edge.

T_{C} is the case temperature, the maximum temperature at the top surface of the component.

$T_{\mathrm{C\,ave}}$ is the average temperature of the top surface of the component.

$T_{\mathrm{B\,max}}$ is the maximum temperature of the printed wiring board.

T_{HSP} is the maximum temperature of the heat spreader.

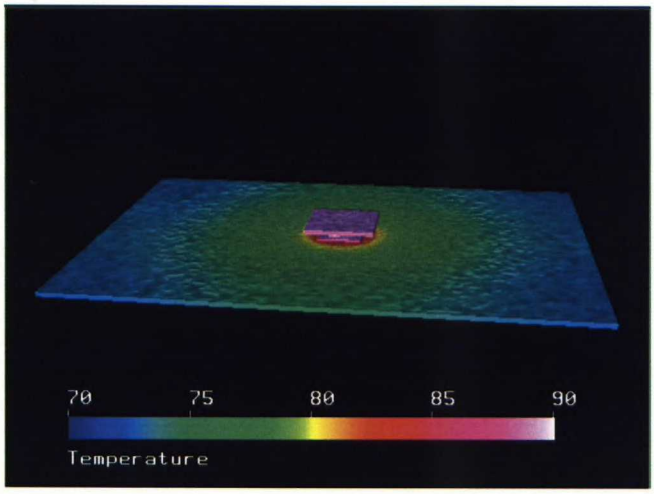

Figure 6.1: Temperature distribution for case 3, variation B. The printed wiring board and heat spreader is shown.

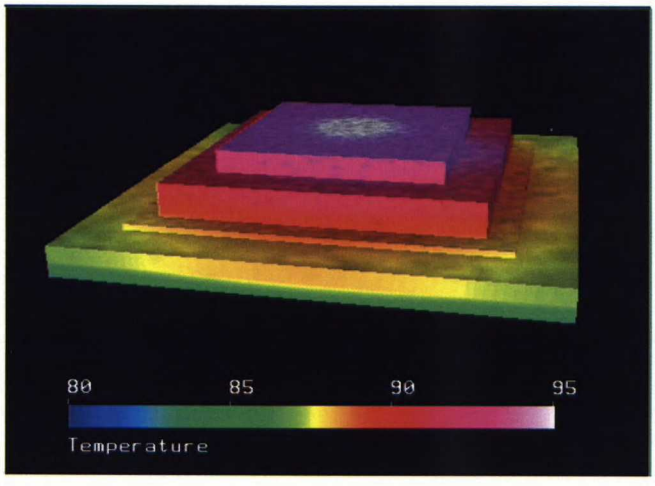

Figure 6.2: Temperature distribution of the chip stack in case 3, variation B. The thermal interface, heat spreader and printed wiring board are omitted.

$T_{\mathrm{HSP\ 1\ mm}}$ is the maximum temperature of the heat spreader at a distance of 1 mm from the component edge.

$\Phi_{\%\mathrm{B}}$ is the heat flux percentage through the bottom of the stack.

$\Phi_{\%\mathrm{T}}$ is the heat flux percentage through the top of the stack.

The heat flux percentages through the top and bottom of the stack are calculated by comparing the heat flux calculated by the `SaveScalars` solver with the total dissipated power of the chip stack. As `SaveScalars` underestimates the heat flux, $\Phi_{\%\mathrm{B}}$ and $\Phi_{\%\mathrm{T}}$ are lower than is really the case. The accuracy of other reference points is higher.

From the reference points and the ambient temperature, T_{Ext}, we get the derived values:

$$\begin{aligned}
\Theta_{\mathrm{JA}} &= (T_{\mathrm{J}} - T_{\mathrm{Ext}})/P_{\mathrm{Tot}}, \\
\Psi_{\mathrm{JB}} &= (T_{\mathrm{J}} - T_{\mathrm{B}})/P_{\mathrm{Tot}}, \\
\Psi_{\mathrm{JT}} &= (T_{\mathrm{J}} - T_{\mathrm{C}})/P_{\mathrm{Tot}}, \\
R_{\mathrm{Th\ JPWB}} &= (T_{\mathrm{J}} - T_{\mathrm{B}})/(\Phi_{\%\mathrm{B}} P_{\mathrm{Tot}}), \\
R_{\mathrm{Th\ JHSP}} &= (T_{\mathrm{J}} - T_{\mathrm{HSP}})/(\Phi_{\%\mathrm{T}} P_{\mathrm{Tot}}).
\end{aligned}$$

Here P_{Tot} is the total power dissipated in the chip stack. Due to mentioned inaccuracy in $\Phi_{\%\mathrm{B}}$ and $\Phi_{\%\mathrm{T}}$, the derived values $R_{\mathrm{Th\ JPWB}}$ and $R_{\mathrm{Th\ JHSP}}$ are $5 - 10$ % higher than those presented in Tabs. A.1, A.2, A.3 and A.4 of the appendix.

The relative error of the maximum temperature when increasing the degrees of freedom in the FEM model is presented in Fig. 6.4. The reference solution is calculated on a mesh with 2.8×10^6 degrees of freedom. Although convergence of the relative error does not guarantee convergence of the solution, convergence of the solution requires convergence of the relative error. For the heat equation, convergence of the relative error typically indicates convergence of the solution.

We have then calculated several further variants of case 3 presented earlier to determine the main effects and interactions of several factors: order of stacking, area of dies, distribution of hotspots, increased convection, and increased thermal conductivity of the thermal interface material, underfill, and inter-chip adhesive.

Nonlinear dependence on the variables shows up as false interaction effects between variables in linear factorial design. As a nonlinear relationship on temperature for several cooling effects is expected, we graphically determine interaction effects. When combining several cooling solutions, the junction temperature will instead asymptotically approach the temperature at the Dirichlet boundary.

The main effects of the variables are presented in Figs. 6.5 and 6.6. Interaction effects are presented in Figs. 6.7 and 6.8. The interaction between

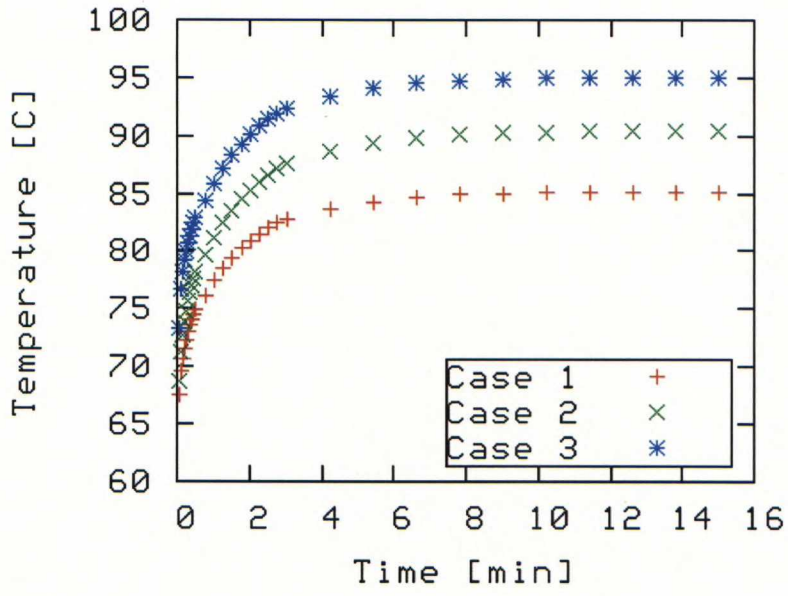

Figure 6.3: Variant A component maximum temperature, corresponding to T_C.

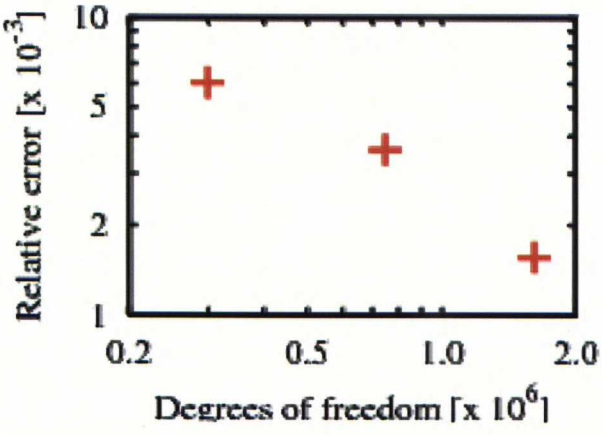

Figure 6.4: The relative error $|\|T\|_{\max} - \|T_R\|_{\max}|/\|T_R\|_{\max}$, where the reference solution T_R is taken to be a finite element solution with 2.8×10^6 degrees of freedom.

the stack order and the die surface area is presented in detail in Fig. 6.9. Table 6.1 summarizes the average maximum case temperature and DRAM junction temperature for the different stack layouts studied. The computed thermal data is presented in Tabs. A.5 through A.8 in the appendix.

The layout of the chip stack together with the conductivity of the thermal interface material layer account for most of the variations in both package top surface temperature and DRAM junction temperature. Significant interaction effects are present for stack order and die surface area. Other interaction effects are present for interaction of the most significant main effects. We suspect that the observed interaction effects are due to the non-linear behavior of the case temperature. The maximum difference in DRAM junction temperature compared to case temperature is 0.75 °C.

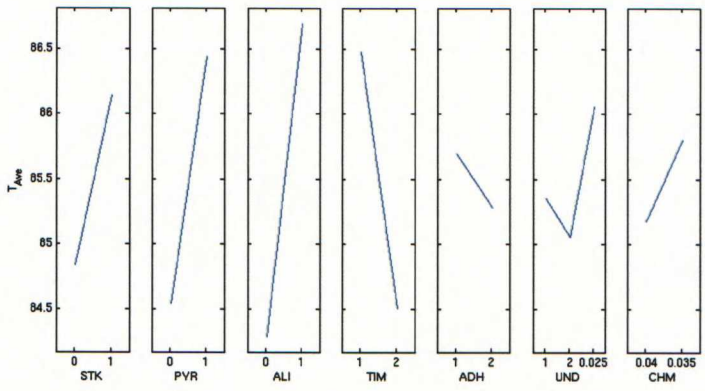

Figure 6.5: Main effects of different factors on maximum case temperature.

From Tab. 6.1 we see that stack design accounts for a temperature difference of up to 6.1 °C in DRAM maximum temperature and 5.6 °C for case temperature.

Figures 6.10-6.12 presents the temperature distribution for two different chip stacks. In Fig. 6.12 the hotspots are distributed, and the temperature distribution is more uniform than compared to the case of centered hotspots presented in Fig. 6.13.

6.2 Thermally induced strain

We calculate the temperature distribution and thermally induced strain in a sample chip stack. The sample chip stack is based on case 3 described in Fig. 5.3. Based on the results presented in Sec. 6.1, we exchange the single DRAM chips for a single four DRAM memory module. We also distribute

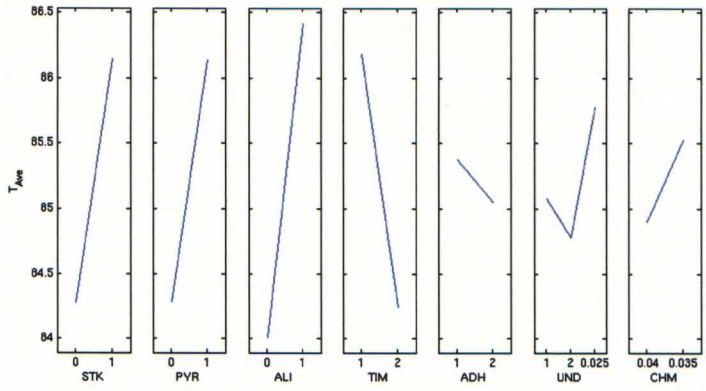

Figure 6.6: Main effects of different factors on maximum DRAM junction temperature.

Table 6.1: Average max case and DRAM junction temperature for different die stack designs.

Design	T_{MAX}	T_{DRAM}
Inverted uniform distributed	83.1	82.6
Inverted uniform aligned	85.7	85.2
Inverted pyramid distributed	84.2	83.6
Inverted pyramid aligned	86.4	85.8
DRAM on top uniform distributed	83.4	83.4
DRAM on top uniform aligned	86.0	86.0
DRAM on top pyramid distributed	86.5	86.5
DRAM on top pyramid aligned	88.7	88.7

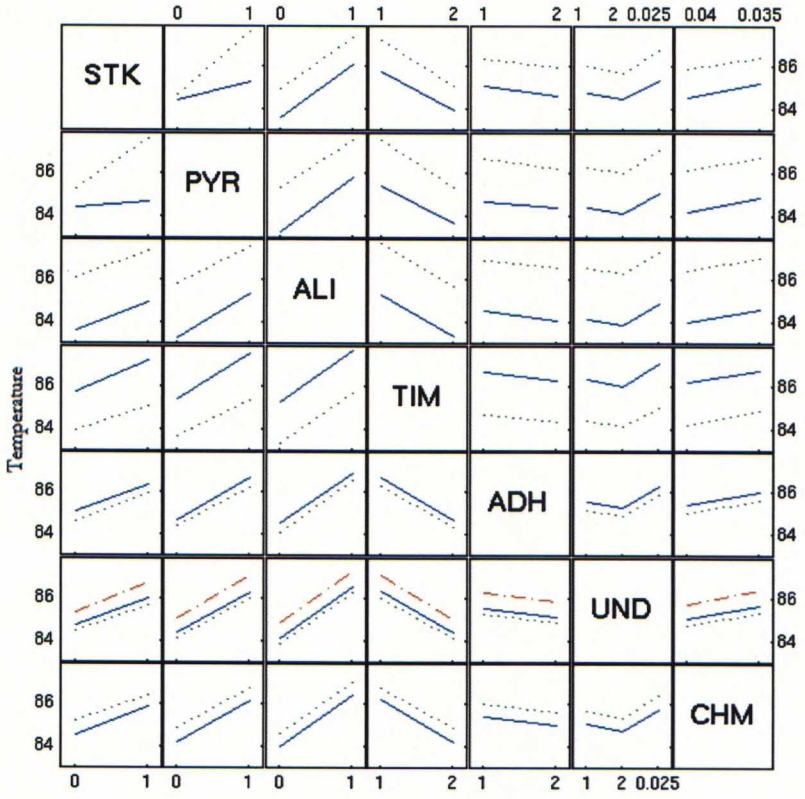

Figure 6.7: Interaction effects of different factors on maximum case temperature. The stack order and die area have significant interaction effects.

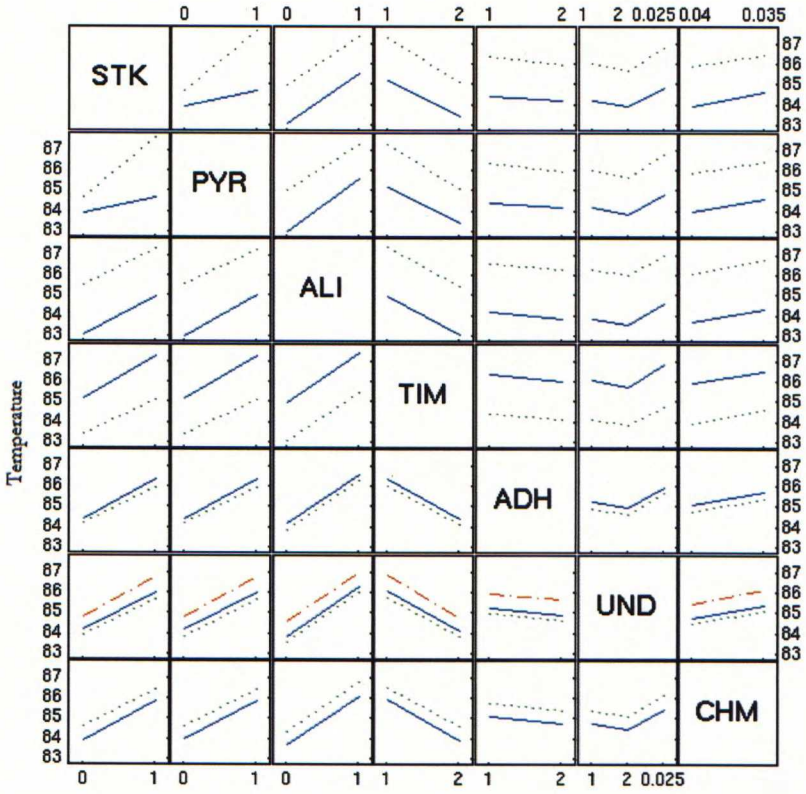

Figure 6.8: Interaction effects of different factors on maximum DRAM junction temperature. The stack order and die area have significant interaction effects.

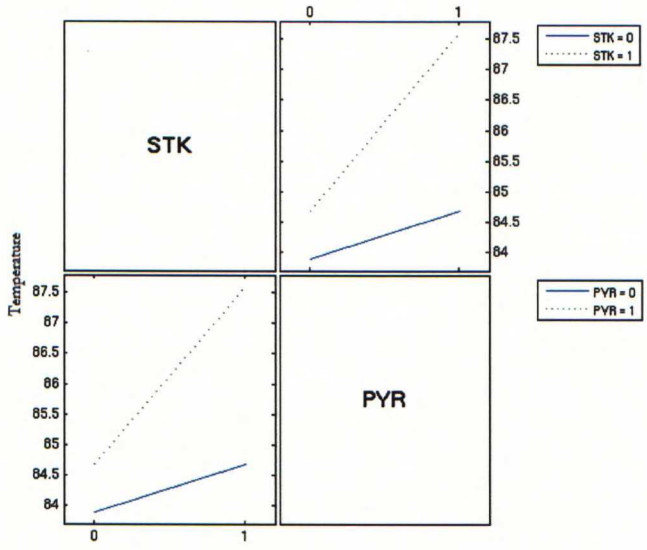

Figure 6.9: Interaction effects of stack order and die surface area for DRAM junction temperature.

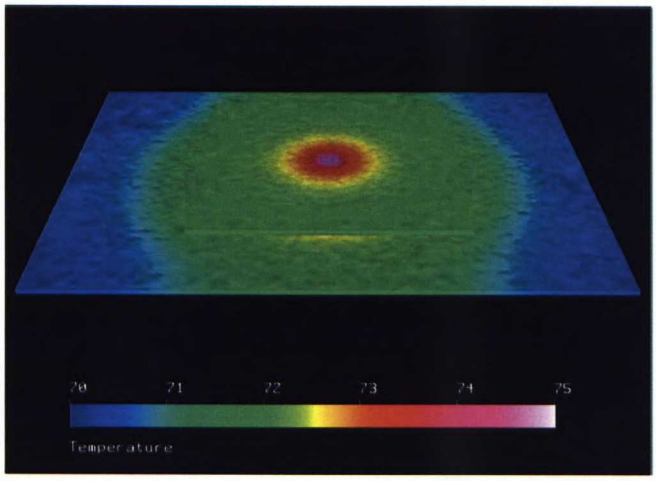

Figure 6.10: Surface temperature of PWB and HSP. Out of scale temperatures are saturated.

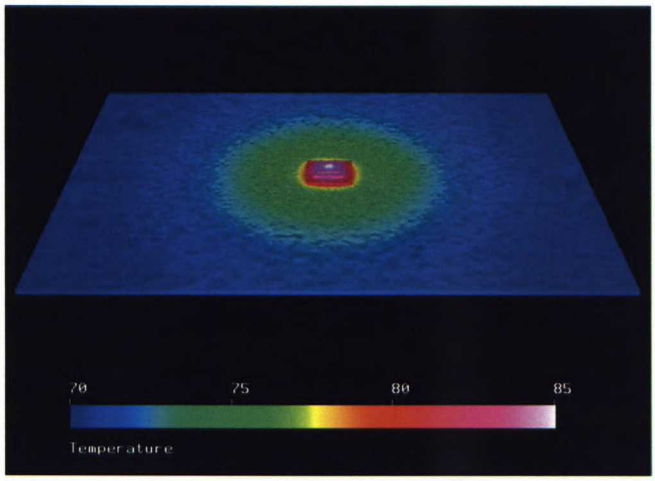

Figure 6.11: Surface temperature of PWB and die stack. Out of scale temperatures are saturated.

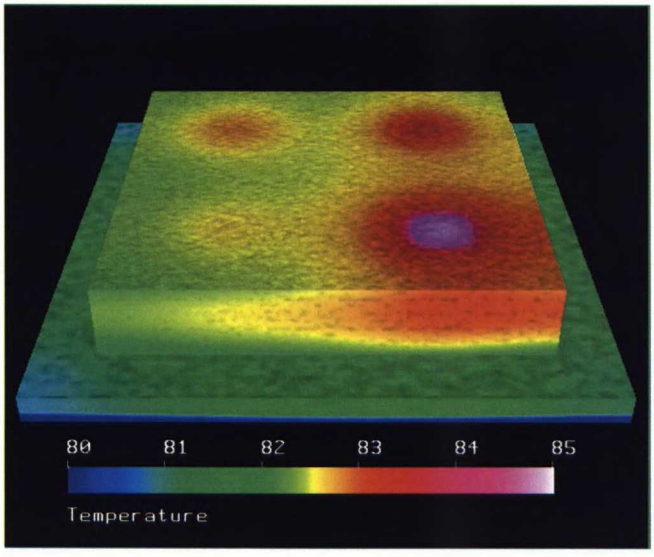

Figure 6.12: Uniform die stack with distributed hot spots. Several lower temperature hotspots are present. Out of scale temperatures are saturated.

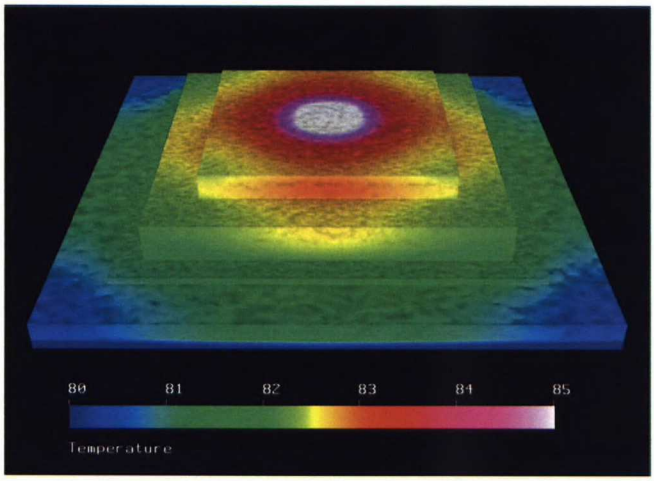

Figure 6.13: Pyramidal die stack with centered hotspots. Note that the temperature of the white area surpasses 85 °C. Out of scale temperatures are saturated.

the thermal hotspots of each die to produce a more uniform temperature distribution.

We expect the thermally induced strain to mainly affect the in stack adhesive interfaces. For this reason, homogenization of the chip and adhesive is no longer possible. Instead we use a structured hexahedral mesh with significantly thinner hexahedrons in the adhesive layer. Due to difficulty in matching a detailed structured mesh of the solder ball array with that required for the adhesive layer, we have chosen to homogenize the solder ball array. Total thermal resistance of the solder ball grid is retained by homogenization of the material. Consequently, thermal stress for the PWB and PWB-stack interface is not reliable. We also constrain the displacement of the basal surface of the PWB. By Saint-Venant's principle, errors in introduced in the chip-PWB interface and at the basal surface of the PWB diminish with distance from these. Thermal strain in inter-chip adhesives should therefore not be significantly affected by these approximations.

The thinnest elements are on the order of 1 μm high, and protrude into the air surrounding the chip stack. Calculating convective heat transfer requires very short time steps for the solution to accurately represent the local velocity and temperature on an element. If we assume that the air moves at a speed of 1 cm/s, time steps on the order of 100 μs are required for the approximate equality $\Delta t = l/v$ to hold. For slower air movement the importance of convection decreases, while faster air movement requires shorter time steps. According to Fig. 6.3, we expect that simulations covering 10–15 minutes to be required before steady state or near steady state is

reached. This would, for 1 cm/s flow, require on the order of 10^7 time steps before steady state conditions are reached. We have chosen not to model convective air movement in the more detailed model.

The temperature distribution of the sample stack is presented in Fig. 6.14. The effect of the thermal resistance of the adhesive layers can be clearly seen in the vertical direction. Parasitic hotspot formation is also clearly seen. The distributed hotspots result in a more uniform temperature distribution and cooler hotspots. Maximum stack temperature does not exceed 85 °C.

The strains induced by the temperature distribution are presented in Figs. 6.15 and 6.16. The largest reliable stresses are at the interface between the TIM and the stack top. The stresses at the solder ball grid array are not reliable. Thermal strains in the HSP are presented in Fig. 6.17, and are significantly smaller than those present in the chip stack.

The number of iterations necessary for convergence of the problem using the BiCGSTAB iterative solver is presented in Fig. 6.18. A solution was obtained using 32 processors on CSC's Murska cluster. The mesh consisted of 1042080 hexahedral elements and 1063063 nodes. The Navier equation was solved for a system with 3189189 degrees of freedom. Resource usage was 218683 s of CPU time, 16950 MB memory, and 6855 s wall time. Resource usage of the multiphysics problem is almost exclusively due to the elastic problem.

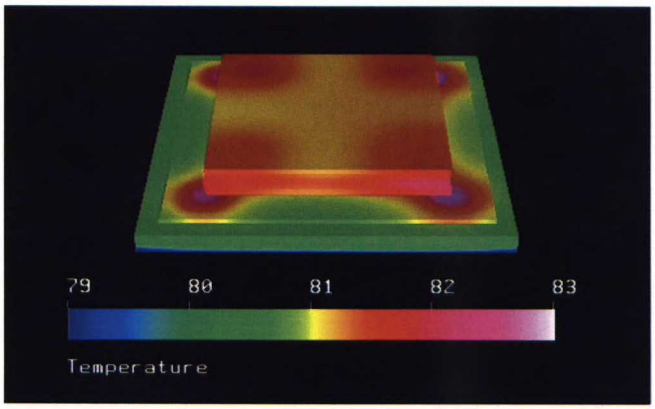

Figure 6.14: Temperature distribution of the stacked chip with four DRAM memory module.

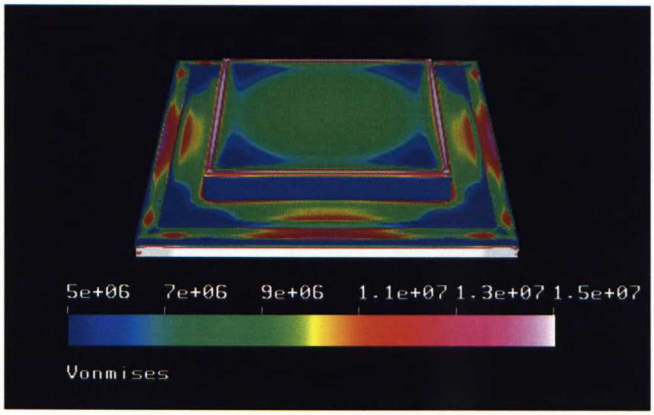

Figure 6.15: Thermal strain in chip stack with four DRAM memory module. In-stack von Mises stress does nowhere exceed 25 MPa

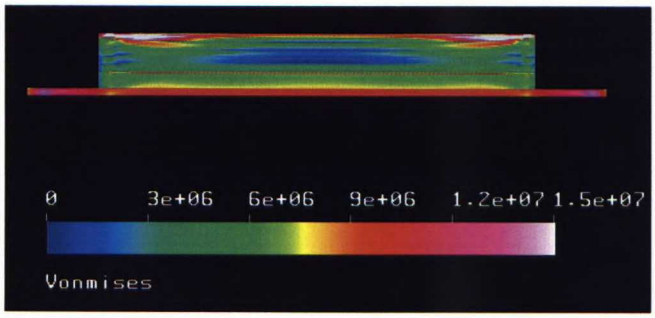

Figure 6.16: Vertical slice of chip stack with multi-DRAM module. The effect of adhesive-silicon layers can be clearly seen.

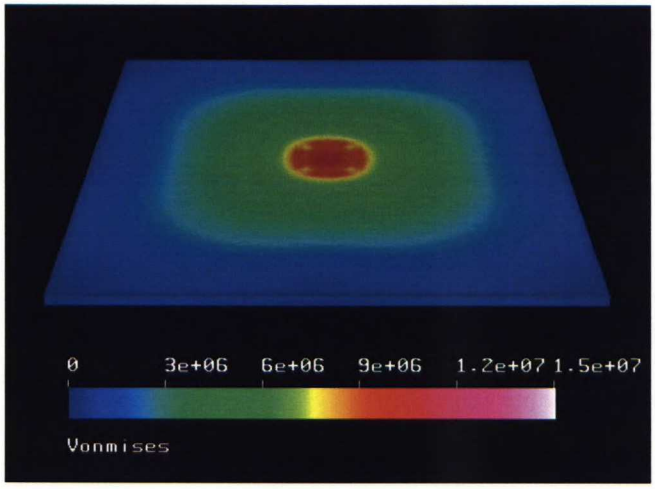

Figure 6.17: Thermal strain in the heat spreader is much smaller than in-stack thermal strain.

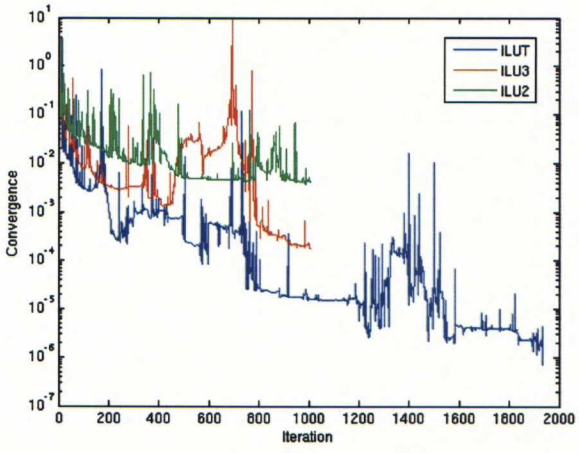

Figure 6.18: Convergence of thermal characterization problem including thermal strains for different preconditioners. Computation done on CSC's Murska cluster using 32 processors.

Chapter 7

Conclusions

In this thesis, we first introduced different methods for handset thermal management. We then described three die stacks and calculated the corresponding temperature distribution. Finally, we calculated thermally induced strains in a chip stack design based on the guidelines obtained.

We have characterized the temperature distribution of three different chip stacks with four variations each. The maximum junction temperature of the memory stack is below 90 °C, while the maximum junction temperatures of the modem stack and the full chip are below 95 °C and 100 °C, respectively. As the temperature increases towards the top of the stack, heat sensitive chips should be placed close to the PWB.

A surprising result is the lower maximum junction temperature of the full chip stack with air underfill compared to epoxy underfill. It was expected that the temperature would increase when the epoxy underfill was replaced with air as it did for the memory and modem stack. In advance, we expected a maximum junction temperature close to 97.5 °C. The calculated maximum junction temperature was, however, 96.03 °C.

The transient simulation of the temperature distribution was only carried out for variant A of cases 1, 2, and 3. The time dependent behavior was entirely as expected. On the other hand, the assumption of a uniform initial temperature distribution is unrealistic, as it requires that no power is dissipated in the entire stack.

Our results indicate that it is feasible to construct 2 W vertical die stacks where hotspot temperatures are below 85 °C. Successful vertical stacking does, however, require consideration of thermal requirements. In the case of limited increase of thermal conductivity in material parameters, it is more efficient to concentrate efficient stack layout.

A larger chip surface provides decreased thermal resistance, and when possible, same-size or near same-size die stacks should be used. The die surface area is especially critical when temperature sensitive components are located at the top of the die stack. Furthermore, it is also advantageous

to horizontally distribute the hotspots of different chips. An inverted stack of dies with off-center hotspots and high power components placed at the bottom of the stack provides the best configuration for vertically stacked dies.

The use of advanced materials in die stacks provides limited cooling benefits, and should be of primary concern only to overcome identified critical thermal bottlenecks. For the computed configuration, a large thermal resistance is present in the thermal interface material between stack and heat-spreader. In-stack adhesive layers provide only limited returns on decreased thermal resistance of the interface layer.

An unexpected result was the observed decrease in case temperature for the inverted pyramid stack. This would not be present for a vertically uniform power dissipation distribution. Instead, the power distribution of the DRAM on top stack is concentrated towards the stack top surface. Inverting the stack brings the main heat sources closer to the wiring board, effectively decreasing the thermal resistance of the package.

Placement of high powered chips close to the board can partly interfere with decreasing the junction temperatures of the DRAM dies. When very high powered components are present, vertical placement of high powered components and DRAM modules should be decided on case by case basis, as minimization of case temperature and memory junction temperature can be partially conflicting objectives.

We have omitted the through vias in all calculations. The simplest way to include their effects would be to modify the Fortran subroutines which calculate the thermal conductance of silicon. In this way it possible to take into account locally increased vertical thermal conductance. A 22×46 copper matrix with a pitch of 20 μm covers an area of 0.5×1.0 mm^2, and increases the thermal conductance of pure silicon by $5 - 10$ %. For a silicon chip of area 5×5 mm^2 this results in a total increase of the thermal conductance by $1 - 2$ %.

Experimental verification of results is limited by prototype production and thermometry. Application specific prototype manufacture must provide reliable in-stack thermometry. The JESD51-4 standard [37] proposes the use of a forward facing PN diode thermometer for this purpose. The sensitivity of the PN diode thermometer is approximately -0.5 °C/mV. Another commonly used thermometer is a thermistor, essentially a temperature sensitive resistor. Due to the bulk generally required in resistors, integration of thermistors is challenging. Thermal surface imaging can also be used independently, or in addition to in-stack thermometry to determine the surface temperature of a die stack.

Thermometry is also possible utilizing quantum mechanical tunneling effects. An electron has a finite probability to tunnel through a classically impassable potential wall. When applying a bias voltage across the junction, a net current across the junction is produced. The current is temperature

dependent, and can consequently be used to determine the temperature of the device. The relative temperature error in this application is expected to be less than 0.02 [38].

A tunnel junction can be manufactured by separating two metal leads by a thin insulating metal-oxide barrier. Aluminum is especially suitable for this purpose as it forms a natural oxide and is suitable for cleanroom use. When not operating as a thermometer an aluminum tunnel junction thermometer can also be used in series to provide sufficient power dissipation for thermal prototypes.

Application of tunnel junction thermometers at an industrial level is limited by junction parameter drift. During prototype measurement drift can be compensated for as the temperature is only dependent on ratio of the conductances and not on the absolute conductance. Drift or hysteresis of the tunnel junction should be negligible as long as junction temperature does not approach 725 K [39].

Bibliography

[1] Franssila, S. Private communication.

[2] Singhal, V., Garimella, S. V. & Raman, A. Microscale pumping technologies for microchannel cooling systems. *ASME Appl. Mech. Rev.* **57**, 191 (2004).

[3] Mohapatra, S. C. & Loikits, D. Advances in liquid coolant technologies for electronics cooling. In *21st IEEE SEMI-THERM Symposium*, 354 (2005).

[4] Garimella, S. V., Singhal, V. & Liu, D. On-chip thermal management with microchannel heat sinks and integrated micropumps. *Proc. IEEE* **94**, 1534 (2006).

[5] Colgan, E. G. *et al.* A practical implementation of silicon microchannel coolers for high power chips. *IEEE Trans. Comp. Packag. Tech.* **30**, 218 (2007).

[6] Prasher, R. S. *et al.* Nano and micro technology-based next-generation package-level cooling solutions. *Intel Tech. J.* **9**, 285 (2005).

[7] Li, H., Jacob, K. I. & Wong, C. P. An improvement of thermal conductivity of underfill materials for flip-chip packages. *IEEE Trans. Adv. Packag.* **26**, 25 (2003).

[8] Prasher, R. S. Rheology based modeling and design of particle laden polymeric thermal interface materials. *IEEE Trans. Comp. Adv. Packag. Tech.* **28**, 230 (2005).

[9] Prasher, R. S. Thermal interface materials: Historical perspective, status, and future directions. *Proc. IEEE* **94**, 1571 (2006).

[10] Schacht, R. *et al.* Characterization of thermal interface materials. In *Electronics Systemintegration Technology Conference*, 1367 (2006).

[11] Geim, A. K. & Novoselov, K. S. The rise of graphene. *Nature Materials* **6**, 183 (2007).

[12] Xu, Y., Zhang, Y., Suhir, E. & Wang, X. Thermal properties of carbon nanotube array used for integrated circuit cooling. *J. Appl. Phys.* **100**, 074302 (2006).

[13] Kordás, K. *et al.* Chip cooling with integrated carbon nanotube microfin architectures. *Appl. Phys. Lett.* **90**, 123105 (2007).

[14] Gong, Z.-X. & Mujumdar, A. S. Cyclic heat transfer in a novel storage unit of multiple phase change materials. *Appl. Therm. Eng.* **16**, 807 (1996).

[15] Kandasamy, R., Wang, X.-Q. & Mujumdar, A. S. Application of phase change materials in thermal management of electronics. *Appl. Therm. Eng.* **27**, 2822 (2007).

[16] Hämäläinen, J. & Järvinen, J. *Elementtimenetelmä virtauslaskennassa* (CSC - Tieteellinen laskenta Oy, 2006).

[17] Feng, K. & Shi, Z.-C. *Mathematical Theory of Elastic Structures* (Springer-Verlag, 1996).

[18] Larsson, S. & Thomée, V. *Partial Differential Equations with Numerical Methods* (Springer, 2005).

[19] Eich-Soellner, E. & Führer, C. *Numerical Methods in Multibody Dynamics* (Lund, 2002).

[20] CSC, the Finnish IT Center for Science. *Elmer Solver Manual* (2007). Http://www.csc.fi/elmer.

[21] Trefethen, L. N. & Bau, D., III. *Numerical Linear Algebra* (SIAM, 1997).

[22] van der Vorst, H. A. Bi-CGSTAB: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. and Stat. Comput.* **13**, 631 (1992).

[23] Saad, Y. *Iterative Methods for Sparse Linear Systems* (PWS Publishing Company, 1996).

[24] Lee, C. C., Palisoc, A. L. & Min, Y. J. Thermal analysis of integrated circuit devices and packages. *IEEE Trans. Comp. Hybrid Manufac. Tech.* **12**, 701 (1989).

[25] De Moerlose, J. & Temmerman, W. Thermal analysis of a chip on board (cob). In *13th IEEE SEMI-THERM Symposium*, 112 (1997).

[26] Chambers, B., Lee, T.-Y. T. & Blood, W. Steady state and transient thermal analysis of chip scale packages. In *1998 InterSociety Conference on Thermal Phenomena* [40], 68.

[27] Pang, J. H. L. & Tan, C.-K. Thermal analysis of a wirebonded chip-on-board package. In *1998 InterSociety Conference on Thermal Phenomena* [40], 481.

[28] Pinel, S. *et al.* Thermal modeling and managemant in ultrathin chip stack technology. *IEEE Trans. Comp. Packag. Technol.* **25**, 244 (2002).

[29] Monfraix, P. *et al.* 3D packaging for space application: imagination and reality. In *EGAAS 2005*, 173 (2005).

[30] The engineering tool box (2007). Http://www.engineeringtoolbox.com.

[31] CSC, the Finnish IT Center for Science. *Elmer Models Manual* (2007). Http://www.csc.fi/elmer.

[32] Geuzaine, C. & Remacle, J.-F. *Gmsh Reference manual* (2007). Http://www.geuz.org/gmsh.

[33] Schöberl, J. *Netgen* (2003). Http://www.hpfem.jku.at/netgen.

[34] Si, H. *TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator* (2006). Http://tetgen.berlios.de.

[35] Karypis, G. & Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comp.* **20**, 359 (1999).

[36] Montgomery, D. C. *Design and Analysis of Experiments* (John Wiley & Sons Inc, 1996), 4th edn.

[37] Electronic Industries Association. *EIA/JESD51: Methodology for the Thermal Measurement of Component Packages* (1995). Http://www.jedec.org.

[38] Gloos, K., Poikolainen, R. S. & Pekola, J. P. Wide-range thermometer based on the temperature-dependent conductance of planar tunnel juntions. *Appl. Phys. Lett.* **77**, 2915 (2000).

[39] Pekola, J. & Meschke, M. Private communication.

[40] *1998 InterSociety Conference on Thermal Phenomena.*

Appendix A

Numerical results

Table A.1: Reference points and requested information for case 1.

	Variant A	Variant B	Variant C	Variant D
T_{J}, DRAM	85.07	85.45	85.82	83.17
T_{J}, Proc. Engine	84.50	84.66	85.24	82.62
T_{J}, Interposer	84.60	84.66	85.32	82.69
T_{C}	85.07	85.45	85.82	83.17
$T_{\mathrm{C\ ave}}$	82.87	83.00	83.65	81.02
$T_{\mathrm{B\ max}}$	82.27	82.19	82.57	80.48
T_{B}	78.28	78.25	78.17	77.00
T_{HSP}	81.80	81.91	82.51	78.74
$T_{\mathrm{HSP\ 1\ mm}}$	81.66	81.77	82.36	78.41
$\Phi_{\%\mathrm{B}}$	88.3 %	88.3 %	87.0 %	77.3 %
$\Phi_{\%\mathrm{T}}$	8.8 %	8.7 %	9.2 %	13.7 %
Θ_{JA}, DRAM	15.48	15.71	15.94	14.30
Ψ_{JB}, DRAM	4.19	4.44	4.72	3.81
Ψ_{JT}, DRAM	0.00	0.00	0.00	0.00
$R_{\mathrm{Th\ JPWB}}$, DRAM	4.75	5.03	5.43	4.93
$R_{\mathrm{Th\ JHSP}}$, DRAM	22.94	25.12	22.21	19.96
Θ_{JA}, Proc. Engine	15.12	15.22	15.58	13.96
Ψ_{JB}, Proc. Engine	3.84	3.96	4.36	3.47
Ψ_{JT}, Proc. Engine	-0.35	-0.49	-0.36	-0.34
$R_{\mathrm{Th\ JPWB}}$, Proc. Engine	4.35	4.48	5.02	4.49
$R_{\mathrm{Th\ JHSP}}$, Proc. Engine	18.94	19.51	18.32	17.48
Θ_{JA}, Interposer	15.19	15.22	15.63	14.01
Ψ_{JB}, Interposer	3.90	3.96	4.41	3.51
Ψ_{JT}, Interposer	-0.29	-0.49	-0.31	-0.30
$R_{\mathrm{Th\ JPWB}}$, Interposer	4.42	4.48	5.07	4.54
$R_{\mathrm{Th\ JHSP}}$, Interposer	19.64	19.51	18.85	17.80

Table A.2: Reference points and requested information for case 2.

	Variant A	Variant B	Variant C	Variant D
T_J, DRAM	89.56	89.79	90.57	87.10
T_J, Modem	90.35	90.96	91.36	87.87
T_J, Proc. Engine	88.83	88.97	89.82	86.49
T_J, Interposer	89.06	89.10	90.04	86.60
T_C	90.35	90.96	91.36	87.87
$T_{C\ ave}$	87.19	87.37	88.22	84.43
$T_{B\ max}$	86.76	86.66	87.30	84.52
T_B	81.85	81.83	81.73	80.30
T_{HSP}	86.53	86.72	87.52	82.93
$T_{HSP\ 1\ mm}$	86.30	86.48	87.28	82.48
$\Phi_{\%B}$	83.7 %	83.6 %	80.0 %	72.6 %
$\Phi_{\%T}$	8.6 %	8.7 %	8.9 %	10.2 %
Θ_{JA}, DRAM	15.24	15.36	15.76	13.97
Ψ_{JB}, DRAM	3.97	4.10	4.56	3.51
Ψ_{JT}, DRAM	-0.41	-0.60	-0.41	-0.40
$R_{Th\ JPWB}$, DRAM	4.75	4.91	5.70	4.83
$R_{Th\ JHSP}$, DRAM	18.16	18.19	17.66	21.07
Θ_{JA}, Modem	15.64	15.96	16.16	14.37
Ψ_{JB}, Modem	4.38	4.71	4.96	3.90
Ψ_{JT}, Modem	0.00	0.00	0.00	0.00
$R_{Th\ JPWB}$, Modem	5.23	5.63	6.20	5.37
$R_{Th\ JHSP}$, Modem	22.89	25.12	22.24	24.96
Θ_{JA}, Proc. Engine	14.86	14.93	15.37	13.64
Ψ_{JB}, Proc. Engine	3.60	3.68	4.17	3.18
Ψ_{JT}, Proc. Engine	-0.78	-1.03	-0.79	-0.73
$R_{Th\ JPWB}$, Proc. Engine	4.30	4.40	5.21	4.37
$R_{Th\ JHSP}$, Proc. Engine	13.79	13.33	13.32	17.84
Θ_{JA}, Interposer	14.98	15.00	15.48	13.71
Ψ_{JB}, Interposer	3.72	3.75	4.28	3.25
Ψ_{JT}, Interposer	-0.66	-0.96	-0.68	-0.65
$R_{Th\ JPWB}$, Interposer	4.44	4.48	5.35	4.47
$R_{Th\ JHSP}$, Interposer	15.16	14.10	14.60	18.55

Table A.3: Reference points and requested information for case 3.

	Variant A	Variant B	Variant C	Variant D
T_J, DRAM0	94.93	96.78	96.03	91.94
T_J, DRAM1	94.76	96.49	95.84	91.82
T_J, DRAM2	94.59	96.27	95.67	91.71
T_J, DRAM3	94.38	96.01	95.47	91.57
T_J, Modem	94.07	95.64	95.17	91.33
T_J, GFX	93.19	94.42	94.30	90.57
T_J, IV/ISP	93.19	94.42	94.30	90.57
T_J, uP	92.92	94.02	94.03	90.34
T_J, N & S	92.32	93.17	93.42	89.77
T_J, NAND	91.35	91.76	92.42	88.82
T_C	94.93	96.78	96.03	91.94
$T_{C\ ave}$	91.50	93.29	92.69	88.66
$T_{B\ max}$	88.91	89.11	89.52	86.54
T_B	82.51	83.07	83.05	81.11
T_{HSP}	87.40	88.76	88.29	78,85
$T_{HSP\ 1\ mm}$	87.17	88.51	88.05	78.32
$\Phi_{\%B}$	85.8 %	84.4 %	82.0 %	74.4 %
$\Phi_{\%T}$	4.9 %	5.3 %	5.4 %	7.5 %
Θ_{JA}, DRAM0	17.47	18.39	18.02	15.97
Ψ_{JB}, DRAM0	6.21	6.86	6.49	5.42
Ψ_{JT}, DRAM0	0.00	0.00	0.00	0.00
$R_{Th\ JPWB}$, DRAM0	7.24	8.12	7.91	7.28
$R_{Th\ JHSP}$, DRAM0	76.84	75.66	71.67	87.27
Θ_{JA}, DRAM1	17.38	18.25	17.92	15.91
Ψ_{JB}, DRAM1	6.13	6.71	6.40	5.36
Ψ_{JT}, DRAM1	-0.09	-0.15	-0.10	-0.06
$R_{Th\ JPWB}$, DRAM1	7.14	7.95	7.80	7.20
$R_{Th\ JHSP}$, DRAM1	75.10	72.92	69.91	84.47
Θ_{JA}, DRAM2	17.30	18.14	17.84	15.86
Ψ_{JB}, DRAM2	6.04	6.60	6.31	5.30
Ψ_{JT}, DRAM2	-0.17	-0.26	-0.18	-0.12
$R_{Th\ JPWB}$, DRAM2	7.04	7.82	7.70	7.12
$R_{Th\ JHSP}$, DRAM2	73.37	70.85	68.33	85.73
Θ_{JA}, DRAM3	17.19	18.01	17.74	15.79
Ψ_{JB}, DRAM3	5.94	6.47	6.21	5.23
Ψ_{JT}, DRAM3	-0.28	-0.39	-0.28	-0.19
$R_{Th\ JPWB}$, DRAM3	6.92	7.67	7.57	7.03
$R_{Th\ JHSP}$, DRAM3	71.22	68.40	66.48	84.80

Table A.4: Additional requested information for case 3.

	Variant A	Variant B	Variant C	Variant D
Θ_{JA}, Modem	17.04	17.82	17.59	15.67
Ψ_{JB}, Modem	5.78	6.29	6.06	5.11
Ψ_{JT}, Modem	-0.43	-0.57	-0.43	-0.31
$R_{Th\ JPWB}$, Modem	6.74	7.44	7.39	6.86
$R_{Th\ JHSP}$, Modem	68.06	64.91	63.70	83.2
Θ_{JA}, GFX, IV/ISP	16.60	17.21	17.15	15.29
Ψ_{JB}, GFX, IV/ISP	5.34	5.68	5.63	4.73
Ψ_{JT}, GFX, IV/ISP	-0.87	-1.18	-0.87	-0.69
$R_{Th\ JPWB}$, GFX, IV/ISP	6.22	6.72	6.86	6.36
$R_{Th\ JHSP}$, GFX, IV/ISP	59.08	53.40	55.65	78.13
Θ_{JA}, uP	16.46	17.01	17.02	15.17
Ψ_{JB}, uP	5.21	5.48	5.49	4.62
Ψ_{JT}, uP	-1.01	-1.38	-1.00	-0.80
$R_{Th\ JPWB}$, uP	6.07	6.49	6.70	6.20
$R_{Th\ JHSP}$, uP	56.33	49.62	53.15	76.6
Θ_{JA}, N & S	16.16	16.59	16.71	14.89
Ψ_{JB}, N & S	4.91	5.05	5.19	4.33
Ψ_{JT}, N & S	-1.31	-1.81	-1.31	-1.09
$R_{Th\ JPWB}$, N & S	5.72	5.98	6.32	5.82
$R_{Th\ JHSP}$, N & S	50.20	41.60	47.50	72.80
Θ_{JA}, NAND	15.68	15.88	16.21	14.41
Ψ_{JB}, NAND	4.42	4.35	4.69	3.86
Ψ_{JT}, NAND	-1.79	-2.51	-1.81	-1.56
$R_{Th\ JPWB}$, NAND	5.15	5.15	5.71	5.18
$R_{Th\ JHSP}$, NAND	40.31	28.30	38.24	66.47

Table A.5: Computed max case and DRAM junction temperature for inverted uniform die size packages with distributed heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
0	0	0	1.0	1.0	0.025	0.035	85.00	84.38
0	0	0	1.0	1.0	0.025	0.04	84.38	83.77
0	0	0	1.0	1.0	1.0	0.035	84.25	83.63
0	0	0	1.0	1.0	1.0	0.04	83.65	83.04
0	0	0	1.0	1.0	2.0	0.035	83.92	83.30
0	0	0	1.0	1.0	2.0	0.04	83.34	82.72
0	0	0	1.0	2.0	0.025	0.035	84.66	84.25
0	0	0	1.0	2.0	0.025	0.04	84.05	83.64
0	0	0	1.0	2.0	1.0	0.035	83.90	83.49
0	0	0	1.0	2.0	1.0	0.04	83.31	82.90
0	0	0	1.0	2.0	2.0	0.035	83.58	83.16
0	0	0	1.0	2.0	2.0	0.04	83.00	82.59
0	0	0	2.0	1.0	0.025	0.035	83.28	82.69
0	0	0	2.0	1.0	0.025	0.04	82.56	81.97
0	0	0	2.0	1.0	1.0	0.035	82.67	82.07
0	0	0	2.0	1.0	1.0	0.04	81.98	81.39
0	0	0	2.0	1.0	2.0	0.035	82.41	81.81
0	0	0	2.0	1.0	2.0	0.04	81.73	81.13
0	0	0	2.0	2.0	0.025	0.035	82.96	82.56
0	0	0	2.0	2.0	0.025	0.04	82.24	81.84
0	0	0	2.0	2.0	1.0	0.035	82.35	81.94
0	0	0	2.0	2.0	1.0	0.04	81.66	81.26
0	0	0	2.0	2.0	2.0	0.035	82.08	81.68
0	0	0	2.0	2.0	2.0	0.04	81.40	81.00

Table A.6: Computed case and max DRAM junction temperature for inverted uniform packages with aligned heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
0	0	1	1.0	1.0	0.025	0.035	87.61	87.01
0	0	1	1.0	1.0	0.025	0.04	86.98	86.38
0	0	1	1.0	1.0	1.0	0.035	86.87	86.26
0	0	1	1.0	1.0	1.0	0.04	86.27	85.66
0	0	1	1.0	1.0	2.0	0.035	86.56	85.94
0	0	1	1.0	1.0	2.0	0.04	85.96	85.36
0	0	1	1.0	2.0	0.025	0.035	87.28	86.81
0	0	1	1.0	2.0	0.025	0.04	86.65	86.19
0	0	1	1.0	2.0	1.0	0.035	86.54	86.06
0	0	1	1.0	2.0	1.0	0.04	85.94	85.47
0	0	1	1.0	2.0	2.0	0.035	86.22	85.74
0	0	1	1.0	2.0	2.0	0.04	85.63	85.16
0	0	1	2.0	1.0	0.025	0.035	85.83	85.28
0	0	1	2.0	1.0	0.025	0.04	85.09	84.55
0	0	1	2.0	1.0	1.0	0.035	85.23	84.68
0	0	1	2.0	1.0	1.0	0.04	84.53	83.98
0	0	1	2.0	1.0	2.0	0.035	84.98	84.42
0	0	1	2.0	1.0	2.0	0.04	84.29	83.73
0	0	1	2.0	2.0	0.025	0.035	85.52	85.08
0	0	1	2.0	2.0	0.025	0.04	84.79	84.36
0	0	1	2.0	2.0	1.0	0.035	84.92	84.48
0	0	1	2.0	2.0	1.0	0.04	84.22	83.78
0	0	1	2.0	2.0	2.0	0.035	84.66	84.22
0	0	1	2.0	2.0	2.0	0.04	83.97	83.54

Table A.7: Computed case and max DRAM junction temperature for inverted pyramid packages with distributed heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
0	1	0	1.0	1.0	0.025	0.035	86.27	85.54
0	1	0	1.0	1.0	0.025	0.04	85.61	84.88
0	1	0	1.0	1.0	1.0	0.035	85.60	84.86
0	1	0	1.0	1.0	1.0	0.04	84.96	84.23
0	1	0	1.0	1.0	2.0	0.035	85.32	84.58
0	1	0	1.0	1.0	2.0	0.04	84.69	83.95
0	1	0	1.0	2.0	0.025	0.035	85.68	85.17
0	1	0	1.0	2.0	0.025	0.04	85.03	84.52
0	1	0	1.0	2.0	1.0	0.035	85.00	84.48
0	1	0	1.0	2.0	1.0	0.04	84.37	83.85
0	1	0	1.0	2.0	2.0	0.035	84.71	84.18
0	1	0	1.0	2.0	2.0	0.04	84.09	83.57
0	1	0	2.0	1.0	0.025	0.035	84.29	83.61
0	1	0	2.0	1.0	0.025	0.04	83.52	82.85
0	1	0	2.0	1.0	1.0	0.035	83.76	83.06
0	1	0	2.0	1.0	1.0	0.04	83.01	82.33
0	1	0	2.0	1.0	2.0	0.035	83.53	82.83
0	1	0	2.0	1.0	2.0	0.04	82.79	82.11
0	1	0	2.0	2.0	0.025	0.035	83.76	83.28
0	1	0	2.0	2.0	0.025	0.04	83.00	82.53
0	1	0	2.0	2.0	1.0	0.035	83.22	82.73
0	1	0	2.0	2.0	1.0	0.04	82.48	82.00
0	1	0	2.0	2.0	2.0	0.035	82.98	82.49
0	1	0	2.0	2.0	2.0	0.04	82.26	81.78

Table A.8: Computed case and max DRAM junction temperature for inverted pyramid packages with aligned heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
0	1	1	1.0	1.0	0.025	0.035	88.49	87.76
0	1	1	1.0	1.0	0.025	0.04	87.82	87.11
0	1	1	1.0	1.0	1.0	0.035	87.81	87.07
0	1	1	1.0	1.0	1.0	0.04	87.17	86.44
0	1	1	1.0	1.0	2.0	0.035	87.53	86.78
0	1	1	1.0	1.0	2.0	0.04	86.89	86.16
0	1	1	1.0	2.0	0.025	0.035	87.98	87.44
0	1	1	1.0	2.0	0.025	0.04	87.32	86.79
0	1	1	1.0	2.0	1.0	0.035	87.29	86.74
0	1	1	1.0	2.0	1.0	0.04	86.65	86.12
0	1	1	1.0	2.0	2.0	0.035	86.99	86.44
0	1	1	1.0	2.0	2.0	0.04	86.37	85.83
0	1	1	2.0	1.0	0.025	0.035	86.49	85.86
0	1	1	2.0	1.0	0.025	0.04	85.72	85.10
0	1	1	2.0	1.0	1.0	0.035	85.95	85.30
0	1	1	2.0	1.0	1.0	0.04	85.21	84.57
0	1	1	2.0	1.0	2.0	0.035	85.72	85.06
0	1	1	2.0	1.0	2.0	0.04	84.99	84.34
0	1	1	2.0	2.0	0.025	0.035	86.05	85.57
0	1	1	2.0	2.0	0.025	0.04	85.29	84.81
0	1	1	2.0	2.0	1.0	0.035	85.49	85.00
0	1	1	2.0	2.0	1.0	0.04	84.76	84.27
0	1	1	2.0	2.0	2.0	0.035	85.26	84.76
0	1	1	2.0	2.0	2.0	0.04	84.53	84.04

Table A.9: Computed max case and DRAM junction temperatures for DRAM on top uniform die size packages with distributed heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
1	0	0	1.0	1.0	0.025	0.035	85.39	85.39
1	0	0	1.0	1.0	0.025	0.04	84.76	84.76
1	0	0	1.0	1.0	1.0	0.035	84.66	84.66
1	0	0	1.0	1.0	1.0	0.04	84.06	84.06
1	0	0	1.0	1.0	2.0	0.035	84.35	84.35
1	0	0	1.0	1.0	2.0	0.04	83.76	83.76
1	0	0	1.0	2.0	0.025	0.035	84.99	84.99
1	0	0	1.0	2.0	0.025	0.04	84.37	84.37
1	0	0	1.0	2.0	1.0	0.035	84.25	84.25
1	0	0	1.0	2.0	1.0	0.04	83.65	83.65
1	0	0	1.0	2.0	2.0	0.035	83.93	83.93
1	0	0	1.0	2.0	2.0	0.04	83.34	83.34
1	0	0	2.0	1.0	0.025	0.035	83.52	83.52
1	0	0	2.0	1.0	0.025	0.04	82.78	82.78
1	0	0	2.0	1.0	1.0	0.035	82.94	82.94
1	0	0	2.0	1.0	1.0	0.04	82.23	82.23
1	0	0	2.0	1.0	2.0	0.035	82.69	82.69
1	0	0	2.0	1.0	2.0	0.04	81.99	81.99
1	0	0	2.0	2.0	0.025	0.035	83.20	83.20
1	0	0	2.0	2.0	0.025	0.04	82.47	82.47
1	0	0	2.0	2.0	1.0	0.035	82.61	82.61
1	0	0	2.0	2.0	1.0	0.04	81.91	81.91
1	0	0	2.0	2.0	2.0	0.035	82.35	82.35
1	0	0	2.0	2.0	2.0	0.04	81.66	81.66

Table A.10: Computed max case and DRAM junction temperatures for DRAM on top uniform die size packages with aligned heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
1	0	1	1.0	1.0	0.025	0.035	87.93	87.93
1	0	1	1.0	1.0	0.025	0.04	87.29	87.29
1	0	1	1.0	1.0	1.0	0.035	87.21	87.21
1	0	1	1.0	1.0	1.0	0.04	86.60	86.60
1	0	1	1.0	1.0	2.0	0.035	86.90	86.90
1	0	1	1.0	1.0	2.0	0.04	86.30	86.30
1	0	1	1.0	2.0	0.025	0.035	87.62	87.62
1	0	1	1.0	2.0	0.025	0.04	86.99	86.99
1	0	1	1.0	2.0	1.0	0.035	86.89	86.89
1	0	1	1.0	2.0	1.0	0.04	86.28	86.28
1	0	1	1.0	2.0	2.0	0.035	86.58	86.58
1	0	1	1.0	2.0	2.0	0.04	85.98	85.98
1	0	1	2.0	1.0	0.025	0.035	85.98	85.98
1	0	1	2.0	1.0	0.025	0.04	85.23	85.23
1	0	1	2.0	1.0	1.0	0.035	85.41	85.41
1	0	1	2.0	1.0	1.0	0.04	84.69	84.69
1	0	1	2.0	1.0	2.0	0.035	85.16	85.16
1	0	1	2.0	1.0	2.0	0.04	84.45	84.45
1	0	1	2.0	2.0	0.025	0.035	85.76	85.76
1	0	1	2.0	2.0	0.025	0.04	85.02	85.02
1	0	1	2.0	2.0	1.0	0.035	85.17	85.17
1	0	1	2.0	2.0	1.0	0.04	84.46	84.46
1	0	1	2.0	2.0	2.0	0.035	84.92	84.92
1	0	1	2.0	2.0	2.0	0.04	84.22	84.22

Table A.11: Computed max case and DRAM junction temperatures for DRAM on top pyramid die size packages with distributed heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
1	1	0	1.0	1.0	0.025	0.035	89.04	89.04
1	1	0	1.0	1.0	0.025	0.04	88.56	88.56
1	1	0	1.0	1.0	1.0	0.035	88.05	88.05
1	1	0	1.0	1.0	1.0	0.04	87.59	87.59
1	1	0	1.0	1.0	2.0	0.035	87.63	87.63
1	1	0	1.0	1.0	2.0	0.04	87.18	87.18
1	1	0	1.0	2.0	0.025	0.035	88.44	88.44
1	1	0	1.0	2.0	0.025	0.04	87.98	87.98
1	1	0	1.0	2.0	1.0	0.035	87.43	87.43
1	1	0	1.0	2.0	1.0	0.04	86.98	86.98
1	1	0	1.0	2.0	2.0	0.035	87.00	87.00
1	1	0	1.0	2.0	2.0	0.04	86.56	86.56
1	1	0	2.0	1.0	0.025	0.035	86.46	86.46
1	1	0	2.0	1.0	0.025	0.04	85.89	85.89
1	1	0	2.0	1.0	1.0	0.035	85.65	85.65
1	1	0	2.0	1.0	1.0	0.04	85.10	85.10
1	1	0	2.0	1.0	2.0	0.035	85.30	85.30
1	1	0	2.0	1.0	2.0	0.04	84.76	84.76
1	1	0	2.0	2.0	0.025	0.035	85.97	85.97
1	1	0	2.0	2.0	0.025	0.04	85.40	85.40
1	1	0	2.0	2.0	1.0	0.035	85.15	85.15
1	1	0	2.0	2.0	1.0	0.04	84.60	84.60
1	1	0	2.0	2.0	2.0	0.035	84.82	84.82
1	1	0	2.0	2.0	2.0	0.04	84.27	84.27

Table A.12: Computed max case and DRAM junction temperatures for DRAM on top pyramid die size packages with aligned heaters.

Stack	Pyramid	Align.	k_{TIM}	k_{ADH}	k_{UND}	k_{CHM}	T_{MAX}	T_{DRAM}
1	1	1	1.0	1.0	0.025	0.035	91.22	91.22
1	1	1	1.0	1.0	0.025	0.04	90.74	90.74
1	1	1	1.0	1.0	1.0	0.035	90.23	90.23
1	1	1	1.0	1.0	1.0	0.04	89.77	89.77
1	1	1	1.0	1.0	2.0	0.035	89.82	89.82
1	1	1	1.0	1.0	2.0	0.04	89.36	89.36
1	1	1	1.0	2.0	0.025	0.035	90.80	90.80
1	1	1	1.0	2.0	0.025	0.04	90.33	90.33
1	1	1	1.0	2.0	1.0	0.035	89.79	89.79
1	1	1	1.0	2.0	1.0	0.04	89.34	89.34
1	1	1	1.0	2.0	2.0	0.035	89.36	89.36
1	1	1	1.0	2.0	2.0	0.04	88.92	88.92
1	1	1	2.0	1.0	0.025	0.035	88.55	88.55
1	1	1	2.0	1.0	0.025	0.04	87.96	87.96
1	1	1	2.0	1.0	1.0	0.035	87.75	87.75
1	1	1	2.0	1.0	1.0	0.04	87.19	87.19
1	1	1	2.0	1.0	2.0	0.035	87.42	87.42
1	1	1	2.0	1.0	2.0	0.04	86.86	86.86
1	1	1	2.0	2.0	0.025	0.035	88.20	88.20
1	1	1	2.0	2.0	0.025	0.04	87.62	87.62
1	1	1	2.0	2.0	1.0	0.035	87.41	87.41
1	1	1	2.0	2.0	1.0	0.04	86.84	86.84
1	1	1	2.0	2.0	2.0	0.035	87.08	87.08
1	1	1	2.0	2.0	2.0	0.04	86.52	86.52